Under consideration for publication in Formal Aspects of Computing

Bisimulations in Calculi Modelling Membranes

Roberto Barbuti[1], Andrea Maggiolo-Schettini[1], Paolo Milazzo[1] and Angelo Troina[2]

[1]Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy
[2]Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy

Abstract. Bisimulations are well–established behavioural equivalences that are widely used to study properties of computer science systems. Bisimulations assume the behaviour of systems to be described as labelled transition systems, and properties of a system can be verified by assessing its bisimilarity with a system one knows to enjoy those properties.

In this paper we show how semantics based on labelled transition systems and bisimulations can be defined for two formalisms for the description of biological systems, both capable of describing membrane interactions. These two formalisms are the Calculus of Looping Sequences (CLS) and Brane Calculi, and since they stem from two different approaches (rewrite systems and process calculi) bisimulation appears to be a good candidate as a general verification method.

We introduce CLS and define a labelled semantics and bisimulations for which we prove some congruence results. We show how bisimulations can be used to verify properties by way of two examples: the description of the regulation of lactose degradation in Escherichia coli and the description of the EGF signalling pathway. We recall the PEP calculus (the simplest of Brane Calculi) and its translation into CLS, we define a labelled semantics and some bisimulation congruences for PEP processes, and we prove that bisimilar PEP processes are translated into bisimilar CLS terms.

Keywords: Calculus of Looping Sequences, Brane Calculi, Labelled Semantics, Bisimulations, Bioinformatics.

1. Introduction

A research field that has had a rapid diffusion in the last few years in theoretical computer science is the application of formal methods to the description and the analysis of biological systems. In particular, formal methods for concurrency are obtaining a noticeable success in the context of *systems biology*, that is the branch of Biology devoted to the study of the functionalities of cells considered as systems of interacting components.

Correspondence and offprint requests to: Paolo Milazzo, Dipartimento di Informatica, Università di Pisa, Largo Pontecorvo 3, 56127 Pisa, Italy. e-mail: milazzo@di.unipi.it

Many formalisms originally developed by computer scientists to model interactive systems have been applied to Biology. Among these formalisms, there are Petri Nets [MDN+00], Hybrid Systems [ABI+01], and process calculi [CDP+04, RSS01, PFM+07]. Moreover, some new formalisms have been proposed to describe biomolecular and membrane interactions [Car05, CCD04, DaL04, PrQ05, RPS+04]. These formalisms may replace the many and often ambiguous notations used by biologists, by allowing precise descriptions of the systems under study. Moreover, they allow the development of advanced simulators for biological systems and the verification of properties of these systems by means of tools, such as model checkers, that are practically unknown to biologists.

Behavioural equivalences have been deeply studied in the field of semantics for concurrent processes to compare the behaviour of different systems. In particular, when such equivalences turn out to be congruences, it holds that the behaviour of two equivalent systems should not be distinguishable for any context the systems are put in. Notions of behavioural equivalences could be useful in Biology as well. Just as an example, the behaviour of two molecules, cells or biological systems, could be considered as equivalent when, in the presence of the same molecular complexes (here considered as a context), they give rise to the same set of biochemical interactions. This kind of equivalences can then be used to define properties which have a relevant biological meaning. For example, by using behavioural equivalences one may: a) assess the equivalence of two drugs from the point of view of effects in order to produce the one which is less costly, b) verify the immunity of a cell from a certain class of viruses, c) ensure that a transplanted tissue is not rejected, etc. Moreover, the use of appropriate behavioural equivalences will also allow to group the states of a system which give rise to the same behaviour, thus reducing the complexity of the state space one wants to analyse.

Bisimulations [Mil89] are well–established behavioural equivalences that are now widely used for the verification of properties of computer science systems. Properties can be verified by assessing the bisimilarity of the considered system with a system one knows to enjoy them. Moreover, given the model of a system, bisimulations can be used to consider equivalent simplified models. The aim of this paper is to show how bisimulation congruences can be defined for two formalisms capable of describing biomolecular and membrane interactions, namely the Calculus of Looping Sequences (CLS) and Brane Calculi. As these formalisms stem from two different approaches, rewrite systems and process calculi, bisimulation appears to be a good candidate as a general verification method.

We introduced CLS in [BMM+06a, BMM+06b]. It is based on term rewriting, hence a CLS model consists of a term and a set of rewrite rules. The terms of CLS are constructed by starting from basic constituent elements and composing them by means of operators of sequencing, looping, containment and parallel composition. Sequencing can be used to describe biological entities such as DNA fragments and proteins. Looping allows tying up the ends of a sequence, thus creating a circular sequence of the constituent elements. We assume that the elements of a circular sequence can rotate, and this motivates the terminology of looping sequence. A looping sequence can represent a membrane. The containment operator can be used to represent that an element is inside the membrane, and parallel composition expresses juxtaposition of elements.

A structural congruence relation allows one to consider as equivalent terms that are intended to represent the same biological system. The evolution of a system is described by a set of rewrite rules to be applied to terms. The definition of the rewrite rules depends on the system and the evolution one wants to represent.

Brane Calculi [Car05] are a family of process calculi specialised in the description of membrane activity, and they allow association of processes with membranes. These processes are composed by actions, the execution of which has an effect on the membrane structure. Some examples of actions are phagocytosis (a membrane engulfs another one), exocytosis (a membrane expels another one), and pinocytosis (a new membrane is created inside another one). These three actions are enough to define the simplest of Brane Calculi, namely the phago/exo/pino (PEP) calculus, that is the only calculus of the family we shall consider in this paper. Other actions, such as fusions of membranes and mitosis can be used to define different calculi of the family. Moreover, extensions of Brane Calculi allow describing interactions with molecules and complexes, such as letting them enter and exit membranes.

The simplest kind of semantics one can consider for these calculi is the reduction semantics, that is given in terms of a transition system in which states are terms and transitions correspond to either rewrite rule applications (in the case of CLS) or execution of actions (in the case of Brane Calculi). This kind of semantics does not allow component–wise reasoning on the behaviour of a term, because the behaviour (the semantics) of a composition of terms cannot be inferred from the behaviour (the semantics) of the components.

To allow component–wise reasoning, the semantics of a formalism must not describe only what happens

inside a component of the system, but also what the component could do by interacting with the environment. This is typically obtained in process calculi by defining semantics based on labelled transition systems (LTSs), where a transition denotes an internal action performed by the process, or a potential interaction of the process with some other process in the environment. Moreover, labelled semantics are necessary for the definition of bisimulations, because bisimulations relate systems that are step by step capable of performing the same internal actions and potential interactions with the environment.

For Brane Calculi we can follow a rather standard approach for the definition of a labelled semantics, as their actions are similar to the communications between processes occurring in process calculi such as CCS [Mil89] and the π–calculus [Mil99]. More precisely, since whole membranes are used as parameters of actions, and since membranes are associated with processes in Brane Calculi, we have to follow an approach similar to that of high–order process calculi in the development of the labelled semantics and of the bisimulations [San96].

As regards CLS, we cannot define a labelled semantics exactly as in process calculi, because CLS is based on rewrite rules rather than on actions, hence we have no actions to be used as transition labels. However, since a transition label should describe a potential interaction with the environment, and since interactions in CLS are described by rewrite rules having more than one component in their left hand side, we have that we can use as labels exactly the components that are missing in the term to obtain the left hand side of a rule. In other words, in CLS an interaction with the environment can be described as the context in which the current term should be placed in order to enable the application of a rewrite rule, and this context could be used as transition label. For example, given the rewrite rule $a \mid b \mapsto c$ (describing the interaction of two components a and b resulting in the production of a component c), we have that a term a can perform the transition $a \xrightarrow{\Box \mid b} c$ where $\Box \mid b$ is the context that would enable the application of the rewrite rule ($\Box$ is a placeholder for the current term), and c is the result of the potential application.

CLS and Brane Calculi allow the description of membrane systems at different levels of abstraction. Brane Calculi consider membranes as atomic objects, while CLS terms can allow a finer description by representing membrane components. In [BMM$^+$06a, Mil07] we have given a translation of the PEP calculus into CLS. In this paper we show that bisimulation relations are preserved by the translation, namely that bisimilar processes of the PEP calculus are translated into bisimilar CLS terms.

The main contribution of this paper is the proposal of behavioural equivalences, in particular bisimulations, as a new tool for the study of biological systems. To support our proposal we consider two case studies of real biological systems and we use bisimulations to verify some properties on their CLS models. The two case studies we consider are the regulation of the lactose degradation in Escherichia coli and the EGF signalling pathway. The former is a well–known biological system and we use it to validate the correctness of our approach. The latter is a system with an important role in cancer development, and we use it to show that bisimulations can be used to compare the behaviour of the same system in different conditions. This is particularly important during drug development to study the effects of different drugs or of the same drugs against different viruses or diseases. Also the results on the PEP calculus and on the translation from PEP to CLS aim at supporting our proposal of behavioural equivalences as new verification tools for biological systems. In fact, the result on the preservation of the bisimulation between these two very different formalisms shows that there exists a general notion of behavioural equivalence for biological systems that does not depend on the language used to describe them.

We have chosen the PEP calculus as it is a representative of an approach in the definition of languages for describing membrane systems and allows definition of semantics and equivalences simpler than the ones for more expressive variants. An extended treatment along the same lines could deal with such variants. We leave to this treatment to give examples of descriptions of biological systems and application of equivalences.

In [BMM$^+$06b] we introduced a labelled semantics for CLS, we defined some bisimulation relations and proved congruence results. In this paper we extend the preliminary results in [BMM$^+$06b] by considering the PEP calculus, for which we give a labelled semantics and bisimulations, and we prove congruence results for the latter. Moreover, we prove that bisimilar PEP processes are translated into bisimilar CLS terms.

The paper is organised as follows. In Section 2 we recall CLS syntax and reduction semantics, then we define a labelled semantics and bisimulations. In Section 3 we show how bisimulations can be used to verify properties on two CLS models of real biological systems. In Section 4 we recall the PEP calculus and we define a labelled semantics and bisimulations for the calculus. In Section 5 we recall the encoding of the PEP calculus into CLS and we show that the encoding preserves bisimilarity. In Section 6 we briefly discuss some related works and draw our conclusions.

2. Calculus of Looping Sequences

In this section we introduce the Calculus of Looping Sequences (CLS). It is essentially based on term rewriting, hence a CLS model consists of a term and a set of rewrite rules. The term is intended to represent the structure of the modelled system, and the rewrite rules the events that may cause the system to evolve.

We define the syntax of CLS and its reduction semantics in terms of a transition system in which states are terms and transitions correspond to rewrite rule applications. Subsequently, we define a labelled semantics in which transitions correspond not only to rewrite rule applications, but also to potential applications that could occur if the considered term would be placed inside a suitable context. Finally, we define bisimulation relations based on the labelled semantics and we give some congruence results.

2.1. Syntax and Reduction Semantics

We start with defining the syntax of terms. We assume a possibly infinite alphabet $\mathcal{E}$ of symbols ranged over by $a, b, c, \ldots$.

Definition 1 (Terms). *Terms* T and *Sequences* S of CLS are given by the following grammar:

$$\begin{array}{lcl} T & ::= & S \mid \left(S\right)^L \rfloor T \mid T \mid T \\ S & ::= & \epsilon \mid a \mid S \cdot S \end{array}$$

where a is a generic element of $\mathcal{E}$, and ϵ represents the empty sequence. We denote with $\mathcal{T}$ the infinite set of terms, and with $\mathcal{S}$ the infinite set of sequences.

In CLS we have a sequencing operator $\_ \cdot \_$, a looping operator $\left(\_\right)^L$, a parallel composition operator $\_ \mid \_$ and a containment operator $\_ \rfloor \_$. Sequencing can be used to concatenate elements of the alphabet $\mathcal{E}$. The empty sequence ϵ denotes the concatenation of zero symbols. A term can be either a sequence, or a looping sequence (that is the application of the looping operator to a sequence) containing another term, or the parallel composition of two terms. By definition, looping and containment are always applied together, hence we can consider them as a single binary operator $\left(\_\right)^L \rfloor \_$ which applies to one sequence and one term.

The biological interpretation of the operators is the following: the main entities which occur in cells are DNA and RNA strands, proteins, membranes, and other macro–molecules. DNA strands (and similarly RNA strands) are sequences of nucleic acids, but they can be seen also at a higher level of abstraction as sequences of genes. Proteins are sequences of amino acids which usually have a very complex three–dimensional structure. In a protein there are usually (relatively) few subsequences, called domains, which actually are able to interact with other entities by means of chemical reactions. CLS sequences can model DNA/RNA strands and proteins by describing each gene or each domain with a symbol of the alphabet. Membranes are closed surfaces, often interspersed with proteins, which may contain something. A closed surface can be modelled by a looping sequence. The elements (or the subsequences) of the looping sequence may represent the proteins on the membrane, and by the containment operator it is possible to specify the content of the membrane. Other macro–molecules can be modelled as single alphabet symbols, or as short sequences. Finally, juxtaposition of entities can be described by the parallel composition of their representations.

Brackets can be used to indicate the order of application of the operators, and we assume $\left(\_\right)^L \rfloor \_$ to have precedence over $\_ \mid \_$. In Figure 1 we show some examples of CLS terms and their visual representation.

In CLS we may have syntactically different terms representing the same structure. We now introduce a structural congruence relation to identify such terms.

Definition 2 (Structural Congruence). The structural congruence relations $\equiv_S$ and $\equiv_T$ are the least congruence relations on sequences and on terms, respectively, satisfying the following rules:

$$S_1 \cdot (S_2 \cdot S_3) \equiv_S (S_1 \cdot S_2) \cdot S_3 \qquad S \cdot \epsilon \equiv_S \epsilon \cdot S \equiv_S S$$

$$S_1 \equiv_S S_2 \text{ implies } S_1 \equiv_T S_2 \text{ and } \left(S_1\right)^L \rfloor T \equiv_T \left(S_2\right)^L \rfloor T$$

$$T_1 \mid T_2 \equiv_T T_2 \mid T_1 \qquad T_1 \mid (T_2 \mid T_3) \equiv_T (T_1 \mid T_2) \mid T_3 \qquad T \mid \epsilon \equiv_T T$$

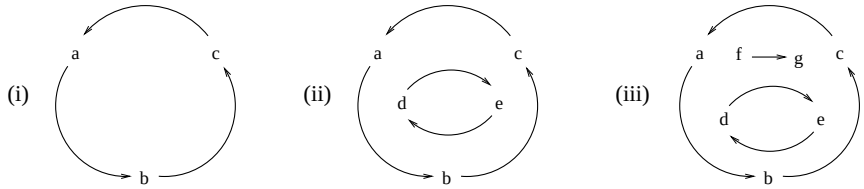

Fig. 1. Visual representation of some examples of CLS terms: (i) represents $\left(a \cdot b \cdot c\right)^L$; (ii) represents $\left(a \cdot b \cdot c\right)^L \rfloor \left(d \cdot e\right)^L$; (iii) represents $\left(a \cdot b \cdot c\right)^L \rfloor \left(\left(d \cdot e\right)^L \mid f \cdot g\right)$.

$$\left(\epsilon\right)^L \rfloor \epsilon \equiv_T \epsilon \qquad \left(S_1 \cdot S_2\right)^L \rfloor T \equiv_T \left(S_2 \cdot S_1\right)^L \rfloor T$$

Rules of the structural congruence state the associativity of $\cdot$ and $\mid$, the commutativity of the latter and the neutral role of ϵ. Moreover, axiom $\left(S_1 \cdot S_2\right)^L \rfloor T \equiv_T \left(S_2 \cdot S_1\right)^L \rfloor T$ says that looping sequences can rotate. In the following, for simplicity, we will use $\equiv$ in place of $\equiv_T$.

Rewrite rules will be defined essentially as pairs of terms, in which the first term describes the portion of the system in which the event modelled by the rule may occur, and the second term describes how that portion of the system changes when the event occurs. In the terms of a rewrite rule we allow the use of variables. As a consequence, a rule will be applicable to all terms which can be obtained by properly instantiating its variables. Variables can be of three kinds: two of these are associated with the two different syntactic categories of terms and sequences, and one is associated with single alphabet elements. We assume a set of term variables TV ranged over by $X, Y, Z, \ldots$, a set of sequence variables SV ranged over by $\widetilde{x}, \widetilde{y}, \widetilde{z}, \ldots$, and a set of element variables $\mathcal{X}$ ranged over by $x, y, z, \ldots$. All these sets are possibly infinite and pairwise disjoint. We denote by $\mathcal{V}$ the set of all variables, $\mathcal{V} = TV \cup SV \cup \mathcal{X}$, and with ρ a generic variable of $\mathcal{V}$. Hence, a pattern is a term which may include variables.

Definition 3 (Patterns). *Patterns* P and *sequence patterns* SP of CLS are given by the following grammar:

$$\begin{array}{rcl} P & ::= & SP \;\mid\; \left(SP\right)^L \rfloor P \;\mid\; P \mid P \;\mid\; X \\ SP & ::= & \epsilon \;\mid\; a \;\mid\; SP \cdot SP \;\mid\; \widetilde{x} \;\mid\; x \end{array}$$

where a is a generic element of $\mathcal{E}$, and $X, \widetilde{x}$ and x are generic elements of TV, SV and $\mathcal{X}$, respectively. We denote with $\mathcal{P}$ the infinite set of patterns.

We assume the structural congruence relation to be trivially extended to patterns. An *instantiation* is a partial function $\sigma : \mathcal{V} \to \mathcal{T}$. An instantiation must preserve the type of variables, thus for $X \in TV, \widetilde{x} \in SV$ and $x \in \mathcal{X}$ we have $\sigma(X) \in \mathcal{T}, \sigma(\widetilde{x}) \in \mathcal{S}$ and $\sigma(x) \in \mathcal{E}$, respectively. Given $P \in \mathcal{P}$, with $P\sigma$ we denote the term obtained by replacing each occurrence of each variable $\rho \in \mathcal{V}$ appearing in P with the corresponding term $\sigma(\rho)$. With Σ we denote the set of all the possible instantiations and, given $P \in \mathcal{P}$, with $Var(P)$ we denote the set of variables appearing in P. Now we define rewrite rules.

Definition 4 (Rewrite Rules). A rewrite rule is a pair of patterns (P_1, P_2), denoted with $P_1 \mapsto P_2$, where $P_1, P_2 \in \mathcal{P}$, $P_1 \not\equiv \epsilon$ and such that $Var(P_2) \subseteq Var(P_1)$. We denote with $\Re$ the infinite set of all the possible rewrite rules.

A rewrite rule $P_1 \mapsto P_2$ states that a term $P_1\sigma$, obtained by instantiating variables in P_1 by some instantiation function σ, can be transformed into the term $P_2\sigma$. We define the reduction semantics of CLS as a transition system, in which states correspond to terms, and transitions correspond to rule applications.

Definition 5 (Reduction Semantics). Given a set of rewrite rules $\mathcal{R} \subseteq \Re$, the *semantics* of CLS is the least transition relation $\to$ on terms closed under $\equiv$, and satisfying the following inference rules:

$$\frac{P_1 \mapsto P_2 \in \mathcal{R} \quad P_1\sigma \not\equiv \epsilon \quad \sigma \in \Sigma}{P_1\sigma \to P_2\sigma} \qquad \frac{T_1 \to T_2}{T \mid T_1 \to T \mid T_2} \qquad \frac{T_1 \to T_2}{\left(S\right)^L \rfloor T_1 \to \left(S\right)^L \rfloor T_2}$$

where the symmetric rule for the parallel composition is omitted.

2.2. Labelled Semantics and Bisimulations

In order to develop the labelled semantics of CLS we have to introduce the notion of *context*, that will be used to define transition labels. The syntax of contexts extends that of terms with a symbol $\Box$ acting as a placeholder for the term that will be placed in the described context.

Definition 6 (Contexts). Contexts $\mathcal{C}$ are given by the following grammar:

$$\mathcal{C} ::= \Box \quad | \quad \mathcal{C} \mid T \quad | \quad T \mid \mathcal{C} \quad | \quad \left(S\right)^{L} \rfloor \mathcal{C}$$

where $T \in \mathcal{T}$ and $S \in \mathcal{S}$. Context $\Box$ is called the *empty context*.

By definition, every context contains a single $\Box$. Let us assume $C, C' \in \mathcal{C}$. With $C[T]$ we denote the term obtained by replacing $\Box$ with T in C; with $C[C']$ we denote the context composition, whose result is the context obtained by replacing $\Box$ with C' in C.

Note that $\left(\Box\right)^{L} \rfloor T$ is not a legal context. The reasons for this are mainly two: the first is that $\Box$ is a placeholder for any term, while the looping operator can be applied only to sequences, the second is that the context must represent the external environment of the term to which it is applied, while $\left(\Box\right)^{L} \rfloor T$ would describe what it is contained within it.

Now we define *parallel contexts* that are a subset of contexts needed in the definition of the labelled semantics.

Definition 7 (Parallel contexts). Parallel contexts $\mathcal{C}_P$ are a subset of contexts given by the following grammar, where $T \in \mathcal{T}$:

$$\mathcal{C}_P ::= \Box \quad | \quad \mathcal{C}_P \mid T \quad | \quad T \mid \mathcal{C}_P$$

Contexts and parallel contexts are used in the labelled semantics of CLS. Moreover, given $T_1, T_2 \in \mathcal{T}$, we write $T_1 \sqcap T_2$ if the two terms share some parallel components, namely if $\exists T' \not\equiv \epsilon, T'', T''' \in \mathcal{T}. T_1 \equiv T' \mid T'' \wedge T_2 \equiv T' \mid T'''$. We write $T_1 \not\sqcap T_2$ otherwise. We assume the definition of $\sqcap$ to be extended to contexts and parallel contexts in the trivial way.

Definition 8 (Labelled Semantics). Given a set of rewrite rules $\mathcal{R} \subseteq \Re$, the *labelled semantics* of CLS is the labelled transition system given by the following inference rules:

$$\text{(rule\_appl)} \ \frac{P \mapsto P' \in \mathcal{R} \quad C[T''] \equiv P\sigma \quad T'' \not\equiv \epsilon \quad \sigma \in \Sigma \quad C \in \mathcal{C}}{T'' \xrightarrow{C} P'\sigma}$$

$$\text{(cont)} \ \frac{T \xrightarrow{\Box} T'}{T'' \rfloor T \xrightarrow{\Box} T'' \rfloor T'} \qquad \text{(par)} \ \frac{T \xrightarrow{C} T' \quad C \in \mathcal{C}_P \quad C[\epsilon] \not\sqcap T''}{T \mid T'' \xrightarrow{C} T' \mid T''}$$

where the dual version of the (par) rule is omitted.

The labelled semantics is similar to the one in [Sew02] for ground term rewriting. A transition $T \xrightarrow{C} T'$ indicates that the application of the context C to T creates an instance of the left part of a rewrite rule, with target instance T'. Intuitively, the transition makes observable the context C, which, when filled with the term T, can be reduced to T', namely $C[T] \mapsto T'$ is an instance of a rewrite rule.

In the definition of the semantics, rule (rule_appl) describes the (potential) application of a rewrite rule. The premise $T'' \not\equiv \epsilon$ implies that the context C used as label cannot provide completely the left part of the rule. For example, if $a \mid b \mapsto c \in \mathcal{R}$, we have that $a \xrightarrow{\Box \mid b} c$ but $\epsilon \not\xrightarrow{a \mid b}$. Without this requirement the labelled semantics would describe not only interactions that could occur between the components described by the current term and some other components in the environment, but also interactions that occur completely in the environment.

Rule (cont) propagates $\Box$–labelled transitions from the inside of a looping sequence to the outside. Transitions labelled with a non–empty context cannot be propagated as they would describe illegal interactions. For example, if $a \mid b \mapsto c \in \mathcal{R}$ we have that $a \xrightarrow{\Box \mid b} c$, but $\left(d\right)^{L} \rfloor a \not\xrightarrow{\Box \mid b}$ as the rewrite rule cannot be applied to $b \mid \left(d\right)^{L} \rfloor a$ and this would describe an illegal interaction between a and b when they are inside and outside the looping sequence, respectively.

Rule (par) propagates transitions labelled with parallel contexts to parallel components. Differently with respect to [Sew02], we allow observation of the context in the reduction of a subterm of a parallel composition. Namely, if $C[T] \mapsto T'$ is an instance of a rewrite rule, then we have that $T|T'' \xrightarrow{C} T'|T''$ (rule (par)). The context can be propagated if it is a parallel context and under the condition that T'' does not provide part of the context C, that is $C[\epsilon] \not\upharpoonright T''$. In this manner we obtain that the context observed is the minimum necessary to apply a rule. Moreover, if the context is not a parallel context we could describe illegal transitions. For example, if $\left(a\right)^L \rfloor\, b \mapsto c \in \mathcal{R}$ we have that $b \xrightarrow{\left(a\right)^L \rfloor\, \Box} c$, but $b \mid d \xrightarrow{\left(a\right)^L \rfloor\, \Box} \not\longrightarrow$ as the rewrite rule cannot be applied to $\left(a\right)^L \rfloor\, (b \mid d)$.

The following theorem states that the labelled semantics is equivalent to the reduction semantics when the context is empty. The proof is immediate.

Theorem 1. $T \to T' \iff T \xrightarrow{\Box} T'$.

Lemma 1 gives a property of parallel contexts, and Lemma 2 gives the labelled semantics with respect to context composition.

Lemma 1. Given $T, T' \in \mathcal{T}$ and a parallel context $C \in \mathcal{C}_P$, it holds that: $C[T]|T' \equiv C[T|T']$.

Proof. Since $C \in \mathcal{C}_P$ there exists T_C such that $C[T] \equiv T_C|T$, and moreover we have that $(T_C|T)|T' \equiv T_C|(T|T') \equiv C[T|T']$. □

Lemma 2. $T \xrightarrow{C[C']} T' \iff C'[T] \xrightarrow{C} T'$.

Proof. By induction on the depth of the derivation tree of $T \xrightarrow{C[C']} T'$.

- *Base case.* Derivation trees of depth 1 are obtained by rule (rule_appl).
  $T \xrightarrow{C[C']} T' \iff$ there exists $P \mapsto P' \in \mathcal{R}$ such that $P\sigma \equiv C[C'[T]]$ and $P'\sigma \equiv T'$ for some instantiation function $\sigma \iff C'[T] \xrightarrow{C} T'$.
- *Induction step.* We assume that the thesis holds for depth n.
  - (par). We first prove the direction $\Longrightarrow$. Let us assume $T = T_1|T_2$; then $T' = T_1'|T_2$, $T_1 \xrightarrow{C[C']} T_1'$, $C[C'] \in \mathcal{C}_P$ and $C[C'][\epsilon] \not\upharpoonright T_2$ (which implies $C[\epsilon] \not\upharpoonright T_2$). We have $C'[T_1] \xrightarrow{C} T_1'$ by induction hypothesis, which implies $C'[T_1]|T_2 \xrightarrow{C} T_1'|T_2$ (by applying rule (par)), and hence $C'[T] \xrightarrow{C} T'$, since $T' = T_1'|T_2$, $C' \in \mathcal{C}_P$ and by Lemma 1. The proof of $\Longleftarrow$ is symmetric.
  - (cont). This case is trivial because $C[C'] = \Box$. □

We introduce a notion of *strong bisimilarity* between CLS terms. This relation compares the behaviour of two terms when the same set of rewrite rules can be applied to both of them. Two terms are strongly bisimilar if at each step of their labelled semantics they are able to perform transitions with the same labels.

Definition 9 (Strong Bisimulation). A binary relation R on terms is a *strong bisimulation* if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

$T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ such that $T_2 \xrightarrow{C} T_2'$ and $T_1' R T_2'$

$T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ such that $T_1 \xrightarrow{C} T_1'$ and $T_1' R T_2'$.

The *strong bisimilarity* $\sim$ is the largest of such relations.

The strong bisimilarity $\sim$ is a congruence with respect to CLS operators, namely, given $T_1, T_2 \in \mathcal{T}$ such that $T_1 \sim T_2$ it holds that $C[T_1] \sim C[T_2]$ for any context $C \in \mathcal{C}$.

Theorem 2 (Strong Congruence). The relation $\sim$ is a congruence.

Proof. The proof that $\sim$ is an equivalence relation is trivial. In order to prove that it is also a congruence we show that $\mathcal{S} \stackrel{def}{=} \{\, (C[T_1], C[T_2]) \mid T_1 \sim T_2 \text{ and } C \text{ is a context} \}$ is a bisimulation. First of all, it is worth noting that $\mathcal{S}$ includes $\sim$ because $C[T_1] = T_1$ when $C = \Box$. Moreover, the following implication holds:

$$T_1 \mathcal{S} T_2 \implies C[T_1] \mathcal{S} C[T_2] \tag{1}$$

because $T_1 \mathcal{S} T_2$ implies $\exists C'.T_1 \equiv C'[T_1'], T_2 \equiv C'[T_2']$ for some $T_1', T_2' \in \mathcal{T}$ such that $T_1' \sim T_2'$. Hence $C[C'[T_1']]\mathcal{S}C[C'[T_2']]$, that is $C[T_1]\mathcal{S}C[T_2]$.

Now, since $\sim$ is a symmetric relation, we have only to show that given $T_1 \sim T_2$ it holds that: $C[T_1] \xrightarrow{C'} T_1' \implies \exists T_2'.C[T_2] \xrightarrow{C'} T_2'$ and $T_1' \mathcal{S} T_2'$.

We prove this by induction on the depth of the derivation tree of $C[T_1] \xrightarrow{C'} T_1'$:

- *Base case* (rule_appl). There exists $P \mapsto P' \in \mathcal{R}$ such that $C'[C[T_1]] \equiv P\sigma$ and $T_1' \equiv P'\sigma$ for some instantiation function σ. This implies $T_1 \xrightarrow{C'[C]} T_1'$ and, since $T_1 \sim T_2$, there exists T_2' such that $T_2 \xrightarrow{C'[C]} T_2'$ with $T_1' \sim T_2'$. Finally, $T_2 \xrightarrow{C'[C]} T_2'$ implies $C[T_2] \xrightarrow{C'} T_2'$ by Lemma 2 and $T_1' \sim T_2'$ implies $T_1' \mathcal{S} T_2'$.
- *Induction step*
  - (par). In this case $C \equiv C_1[C_2]$ for some C_2 and where $C_1 \equiv \Box | T$ for some T. Hence, $C[T_1] \equiv C_1[C_2[T_1]]$ and by the premise of the inference rule we obtain $C_2[T_1] \xrightarrow{C'} T_1''$ with $T \not\frown C'[\epsilon]$. It follows that $T_1' \equiv C_1[T_1'']$. By the induction hypothesis we have that $\exists T_2''.C_2[T_2] \xrightarrow{C'} T_2'' \wedge T_1'' \mathcal{S} T_2''$, hence, by applying the (par) rule, $C_1[C_2[T_2]] \xrightarrow{C'} C_1[T_2'']$, that is $C[T_2] \xrightarrow{C'} T_2'$. By the closure of $\mathcal{S}$ to contexts given in (1), we have $C_1[T_1'']\mathcal{S}C_1[T_2'']$, that is $T_1'\mathcal{S}T_2'$.
  - (cont). In this case $C' = \Box$ and $C \equiv C_1[C_2]$ for some C_2 and where $C_1 \equiv T \rfloor \Box$ for some T. Hence, $C[T_1] \equiv C_1[C_2[T_1]]$ and by the premise of the inference rule we obtain $C_2[T_1] \xrightarrow{\Box} T_1''$. It follows that $T_1' \equiv C_1[T_1'']$. By the induction hypothesis we have that $\exists T_2''.C_2[T_2] \xrightarrow{C'} T_2'' \wedge T_1'' \mathcal{S} T_2''$, hence, by applying the (cont) rule, $C_1[C_2[T_2]] \xrightarrow{\Box} C_1[T_2'']$, that is $C[T_2] \xrightarrow{\Box} T_2'$. By the closure of $\mathcal{S}$ to contexts given in (1), we have $C_1[T_1'']\mathcal{S}C_1[T_2'']$, that is $T_1'\mathcal{S}T_2'$. $\Box$

We denote with $\stackrel{\Box}{\Longrightarrow}$ a sequence of zero or more transitions $\xrightarrow{\Box}$, and with $\stackrel{C}{\Longrightarrow}$, where $C \neq \Box$, the sequence of transitions such that $T \stackrel{C}{\Longrightarrow} T'$ if and only if there exist $T_1, T_2 \in \mathcal{T}$ such that $T \stackrel{\Box}{\Longrightarrow} T_1 \xrightarrow{C} T_2 \stackrel{\Box}{\Longrightarrow} T'$. We have two lemmata.

Lemma 3. If one of the following two conditions holds:

1. $C, C' \in \mathcal{C}_P$ with $C \not\frown C'$,
2. $C = \Box$, $C' \in \mathcal{C}$,

then $T \stackrel{C}{\Longrightarrow} T' \iff C'[T] \stackrel{C}{\Longrightarrow} C'[T']$.

Proof. By definition of $\stackrel{C}{\Longrightarrow}$ and of the labelled semantics. $\Box$

Lemma 4. $T \stackrel{C[C']}{\Longrightarrow} T' \iff C'[T] \stackrel{C}{\Longrightarrow} T'$.

Proof. First of all, it is worth noticing that, by Lemma 3, $T \stackrel{\Box}{\Longrightarrow} T' \iff C[T] \stackrel{\Box}{\Longrightarrow} C[T']$ for any context C. Now, $T \stackrel{C[C']}{\Longrightarrow} T' \iff$ there exist T_1, T_2 such that $T \stackrel{\Box}{\Longrightarrow} T_1 \xrightarrow{C[C']} T_2 \stackrel{\Box}{\Longrightarrow} T'$. By Lemma 2, we have that $C'[T_1] \xrightarrow{C} T_2$, and hence $C'[T] \stackrel{\Box}{\Longrightarrow} C'[T_1] \xrightarrow{C} T_2 \stackrel{\Box}{\Longrightarrow} T'$, that is $C'[T] \stackrel{C}{\Longrightarrow} T'$. $\Box$

Now we introduce a relation called *weak bisimilarity* to compare the behaviour of terms without taking into account system internal moves.

Definition 10 (Weak Bisimulation). A binary relation R on terms is a *weak bisimulation* if, given T_1, T_2 such that $T_1 R T_2$, the two following conditions hold:

$T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ such that $T_2 \stackrel{C}{\Longrightarrow} T_2'$ and $T_1' R T_2'$

$T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ such that $T_1 \stackrel{C}{\Longrightarrow} T_1'$ and $T_1' R T_2'$.

The *weak bisimilarity* $\approx$ is the largest of such relations.

As the strong bisimilarity $\sim$, also the weak bisimilarity $\approx$ is a congruence with respect to CLS operators.

Theorem 3 (Weak Congruence). The relation $\approx$ is a congruence.

Proof. The proof that $\approx$ is an equivalence relation is trivial. In order to prove that it is also a congruence we show that $\mathcal{S} \stackrel{def}{=} \{ (C[T_1], C[T_2]) \mid T_1 \approx T_2 \text{ and } C \text{ is a context}\}$ is a weak bisimulation. Similarly as in the proof of Proposition 2 we have that $\mathcal{S}$ includes $\approx$, and that the following implication holds:

$$T_1 \mathcal{S} T_2 \implies C[T_1] \mathcal{S} C[T_2] \tag{2}$$

and we have only to show that given $T_1 \approx T_2$ it holds that: $C[T_1] \stackrel{C'}{\longrightarrow} T_1' \implies \exists T_2'.C[T_2] \stackrel{C'}{\Longrightarrow} T_2'$ and $T_1' \mathcal{S} T_2'$.

We prove this by induction on the depth of the derivation tree of $C[T_1] \stackrel{C'}{\longrightarrow} T_1'$:

- *Base case* (rule_appl). There exists $P \mapsto P' \in \mathcal{R}$ such that $C'[C[T_1]] \equiv P\sigma$ and $T_1' \equiv P'\sigma$ for some instantiation function σ. This implies $T_1 \stackrel{C'[C]}{\longrightarrow} T_1'$ and, since $T_1 \approx T_2$, there exists T_2' such that $T_2 \stackrel{C'[C]}{\Longrightarrow} T_2'$ with $T_1' \approx T_2'$. Finally, $T_2 \stackrel{C'[C]}{\Longrightarrow} T_2'$ implies $C[T_2] \stackrel{C'}{\Longrightarrow} T_2'$ by Lemma 4 and $T_1' \approx T_2'$ implies $T_1' \mathcal{S} T_2'$.
- *Induction step*
  - (par). In this case $C \equiv C_1[C_2]$ for some C_2 and where $C_1 \equiv \Box | T$ for some T. Hence, $C[T_1] \equiv C_1[C_2[T_1]]$ and by the premise of the inference rule we obtain $C_2[T_1] \stackrel{C'}{\longrightarrow} T_1''$ with $T \not\upharpoonright C'[\epsilon]$. It follows $T_1' \equiv C_1[T_1'']$. By the induction hypothesis we have that $\exists T_2''.C_2[T_2] \stackrel{C'}{\Longrightarrow} T_2'' \wedge T_1'' \mathcal{S} T_2''$, hence, by Lemma 3, $C_1[C_2[T_2]] \stackrel{C'}{\Longrightarrow} C_1[T_2'']$, that is $C[T_2] \stackrel{C'}{\Longrightarrow} T_2'$. By the closure of $\mathcal{S}$ to contexts given in (2), we have $C_1[T_1'] \mathcal{S} C_1[T_2'']$, that is $T_1' \mathcal{S} T_2'$.
  - (cont). In this case $C' = \Box$ and $C \equiv C_1[C_2]$ for some C_2 and where $C_1 \equiv T \rfloor \Box$ for some T. Hence, $C[T_1] \equiv C_1[C_2[T_1]]$ and by the premise of the inference rule we obtain $C_2[T_1] \stackrel{\Box}{\longrightarrow} T_1''$. It follows that $T_1' \equiv C_1[T_1'']$. By the induction hypothesis we have that $\exists T_2''.C_2[T_2] \stackrel{C'}{\Longrightarrow} T_2'' \wedge T_1'' \mathcal{S} T_2''$, hence, by Lemma 3, $C_1[C_2[T_2]] \stackrel{\Box}{\Longrightarrow} C_1[T_2'']$, that is $C[T_2] \stackrel{\Box}{\Longrightarrow} T_2'$. By the closure of $\mathcal{S}$ to contexts given in (2), we have $C_1[T_1''] \mathcal{S} C_1[T_2'']$, that is $T_1' \mathcal{S} T_2'$. $\square$

Now we give a simple example that shows the differences between strong and weak bisimilarities.

Example 1. Consider the following set of rules:

$$\mathcal{R} = \{ \quad a \mid b \mapsto c \quad , \quad d \mid b \mapsto e \quad , \quad e \mapsto e \quad , \quad c \mapsto e \quad , \quad f \mapsto a \quad \}.$$

We have that $a \sim d$, because $a \stackrel{\Box|b}{\longrightarrow} c \stackrel{\Box}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} \ldots$ and $d \stackrel{\Box|b}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} \ldots$, and $f \approx d$, because $f \stackrel{\Box}{\longrightarrow} a \stackrel{\Box|b}{\longrightarrow} c \stackrel{\Box}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} e \stackrel{\Box}{\longrightarrow} \ldots$. On the other hand, $f \not\sim e$ and $f \not\approx e$. $\square$

In the definition of strong and weak bisimilarities we assumed the set of rewrite rules to be applied to the considered terms to be the same. However, one may also be interested in comparing the behaviour of terms whose evolution is given by the application of two possibly different sets of rewrite rules. To this aim we define *CLS systems* as pairs consisting of a CLS term and a set of rewrite rules, and we define bisimulations on CLS systems.

Definition 11 (System). A *CLS System* is a pair $\langle T, \mathcal{R} \rangle$ with $T \in \mathcal{T}$, $\mathcal{R} \subseteq \Re$.

Given a system $\langle T, \mathcal{R} \rangle$, we write $\mathcal{R} : T \stackrel{C}{\longrightarrow} T'$ to mean that the transition $T \stackrel{C}{\longrightarrow} T'$ is performed by applying a rule in $\mathcal{R}$, and we write $\mathcal{R} : T \stackrel{C}{\Longrightarrow} T'$ to mean that the sequence of transitions $T \stackrel{C}{\Longrightarrow} T'$ is performed by applying rules in $\mathcal{R}$. Now, we introduce strong and weak bisimilarities between CLS systems. With abuse of notation we denote such relations with $\sim$ and $\approx$, respectively.

Definition 12 (Strong Bisimulation on Systems). A binary relation R on CLS systems is a *strong bisimulation* if, given $\langle T_1, \mathcal{R}_1 \rangle$ and $\langle T_2, \mathcal{R}_2 \rangle$ such that $\langle T_1, \mathcal{R}_1 \rangle R \langle T_2, \mathcal{R}_2 \rangle$, the two following conditions hold:

$\mathcal{R}_1 : T_1 \stackrel{C}{\longrightarrow} T_1' \implies \exists T_2'$ such that $\mathcal{R}_2 : T_2 \stackrel{C}{\longrightarrow} T_2'$ and $\langle T_1', \mathcal{R}_1 \rangle R \langle T_2', \mathcal{R}_2 \rangle$

$\mathcal{R}_2 : T_2 \stackrel{C}{\longrightarrow} T_2' \implies \exists T_1'$ such that $\mathcal{R}_1 : T_1 \stackrel{C}{\longrightarrow} T_1'$ and $\langle T_1', \mathcal{R}_1 \rangle R \langle T_2', \mathcal{R}_2 \rangle$.

The *strong bisimilarity* $\sim$ is the largest of such relations.

Definition 13 (Weak Bisimulation on Systems). A binary relation R on CLS systems is a *weak bisimulation* if, given $\langle T_1, \mathcal{R}_1\rangle$ and $\langle T_2, \mathcal{R}_2\rangle$ such that $\langle T_1, \mathcal{R}_1\rangle R \langle T_2, \mathcal{R}_2\rangle$, the two following conditions hold:

$\mathcal{R}_1 : T_1 \xrightarrow{C} T_1' \implies \exists T_2'$ such that $\mathcal{R}_2 : T_2 \stackrel{C}{\Longrightarrow} T_2'$ and $\langle T_1', \mathcal{R}_1\rangle R \langle T_2', \mathcal{R}_2\rangle$

$\mathcal{R}_2 : T_2 \xrightarrow{C} T_2' \implies \exists T_1'$ such that $\mathcal{R}_1 : T_1 \stackrel{C}{\Longrightarrow} T_1'$ and $\langle T_1', \mathcal{R}_1\rangle R \langle T_2', \mathcal{R}_2\rangle$.

The *weak bisimilarity* $\approx$ is the largest of such relations.

If we fix a set of rewrite rules, strong and weak bisimilarities on CLS systems correspond to strong and weak bisimilarities on terms, respectively. Namely, for a given $\mathcal{R} \in \Re$, $\langle T_1, \mathcal{R}\rangle \sim \langle T_2, \mathcal{R}\rangle$ if and only if $T_1 \sim T_2$ and $\langle T_1, \mathcal{R}\rangle \approx \langle T_2, \mathcal{R}\rangle$ if and only if $T_1 \approx T_2$. However, as we show in the following example, bisimilarity relations introduced for CLS systems are not congruences, namely $\langle T_1, \mathcal{R}_1\rangle \sim \langle T_2, \mathcal{R}_2\rangle$ and $\langle T_1, \mathcal{R}_1\rangle \approx \langle T_2, \mathcal{R}_2\rangle$ do not imply $\langle C[T_1], \mathcal{R}_1\rangle \sim \langle C[T_2], \mathcal{R}_2\rangle$ and $\langle C[T_1], \mathcal{R}_1\rangle \approx \langle C[T_2], \mathcal{R}_2\rangle$, respectively, for any context $C \in \mathcal{C}$.

Example 2. Let $\mathcal{R}_1 = \{a \mid b \mapsto c\}$ and $\mathcal{R}_2 = \{a \mid d \mapsto c\ ,\ \ b \mid e \mapsto c\}$. We have that $\langle a, \mathcal{R}_1\rangle \sim \langle e, \mathcal{R}_2\rangle$ because $\mathcal{R}_1 : a \xrightarrow{\Box \mid b} c$ and $\mathcal{R}_2 : e \xrightarrow{\Box \mid b} c$. Moreover, $\langle b, \mathcal{R}_1\rangle \sim \langle d, \mathcal{R}_2\rangle$ because $\mathcal{R}_1 : b \xrightarrow{\Box \mid a} c$ and $\mathcal{R}_2 : d \xrightarrow{\Box \mid a} c$. However, $\langle a \mid b, \mathcal{R}_1\rangle \not\sim \langle e \mid d, \mathcal{R}_2\rangle$ because $\mathcal{R}_1 : a \mid b \xrightarrow{\Box} c$, while $\mathcal{R}_2 : d \mid e \not\xrightarrow{\Box} c$. This is a proof that the strong bisimilarity on systems is not a congruence because otherwise we could prove $\langle a \mid b, \mathcal{R}_1\rangle \sim \langle e \mid b, \mathcal{R}_2\rangle$ by composing a and e with the same context $\Box \mid b$, and $\langle e \mid b, \mathcal{R}_1\rangle \sim \langle e \mid d, \mathcal{R}_2\rangle$ by composing b and d with the same context $e \mid \Box$, and hence $\langle a \mid b, \mathcal{R}_1\rangle \sim \langle e \mid d, \mathcal{R}_2\rangle$ by the transitivity of $\sim$. □

Even if bisimilarities on CLS systems are not congruences, they allow us to define equivalence relations on sets of rewrite rules.

Definition 14 (Rules Equivalence). Two sets of rewrite rules $\mathcal{R}_1$ and $\mathcal{R}_2$ are strongly (resp. weakly) equivalent, denoted $\mathcal{R}_1 \simeq \mathcal{R}_2$ (resp. $\mathcal{R}_1 \cong \mathcal{R}_2$), if and only if for any term $T \in \mathcal{T}$ it holds that $\langle T, \mathcal{R}_1\rangle \sim \langle T, \mathcal{R}_2\rangle$ (resp. $\langle T, \mathcal{R}_1\rangle \approx \langle T, \mathcal{R}_2\rangle$).

Example 3. Given $\mathcal{R}_1 = \{a \mapsto c\}$, $\mathcal{R}_2 = \{a \mapsto f\}$ and $\mathcal{R}_3 = \{a \mapsto b\ ,\ \ b \mapsto c\}$, we have that $\mathcal{R}_1 \simeq \mathcal{R}_2$, but $\mathcal{R}_1 \not\simeq \mathcal{R}_3$ and $\mathcal{R}_1 \cong \mathcal{R}_3$. □

Now, if we resort to equivalent rules, we can prove congruence results on CLS systems.

Theorem 4 (Congruences on Systems). Given $\mathcal{R}_1 \simeq \mathcal{R}_2$ (resp. $\mathcal{R}_1 \cong \mathcal{R}_2$) and $\langle T, \mathcal{R}_1\rangle \sim \langle T', \mathcal{R}_2\rangle$ (resp. $\langle T, \mathcal{R}_1\rangle \approx \langle T', \mathcal{R}_2\rangle$), for any $C \in \mathcal{C}$ we have $\langle C[T], \mathcal{R}_1\rangle \sim \langle C[T'], \mathcal{R}_2\rangle$ (resp. $\langle C[T], \mathcal{R}_1\rangle \approx \langle C[T'], \mathcal{R}_2\rangle$).

Proof. Since $\mathcal{R}_1 \simeq \mathcal{R}_2$ we have that $\langle T, \mathcal{R}_1\rangle \sim \langle T, \mathcal{R}_2\rangle$; moreover, by hypothesis, $\langle T, \mathcal{R}_1\rangle \sim \langle T', \mathcal{R}_2\rangle$, and therefore $\langle T, \mathcal{R}_2\rangle \sim \langle T', \mathcal{R}_2\rangle$. Now, since the set of rewrite rules is the same ($\mathcal{R}_2$), by the congruence results for CLS terms, we have $\langle C[T], \mathcal{R}_2\rangle \sim \langle C[T'], \mathcal{R}_2\rangle$. Again, since $\mathcal{R}_1 \simeq \mathcal{R}_2$, we have $\langle C[T], \mathcal{R}_1\rangle \sim \langle C[T], \mathcal{R}_2\rangle$, and hence, $\langle C[T], \mathcal{R}_1\rangle \sim \langle C[T'], \mathcal{R}_2\rangle$. The proof is identical for $\cong$ and $\approx$ instead of $\simeq$ and $\sim$, respectively. □

3. Application to the Modelling of Cellular Pathways

In this section we develop the CLS models of two real biological systems. The first is the well–known regulation process of the lactose operon in E. coli (Escherichia coli), and the second is the EGF signalling pathway, which is a network of protein interactions with an important role in cancer development.

The aim of the first case study is to validate the correctness of our approach, in fact we shall use the weak bisimulation on systems to prove a property implying that the model correctly reproduces the expected behaviour of the system. As regards the second case study, the aim is to show the applicability of our approach to systems biology by employing the weak bisimulation on terms to study how the behaviour of the described system changes in different situations. This is particularly important during drug development to compare the effects of different drugs or of the same drug against different viruses or diseases.

From a technical point of view, the first case study shows how to modify rewrite rules of a model in order to observe an internal event from outside, whereas the second case study shows that bisimulations on systems can be used as a tool to prove the equivalence of two models with respect to the corresponding bisimulations on terms.

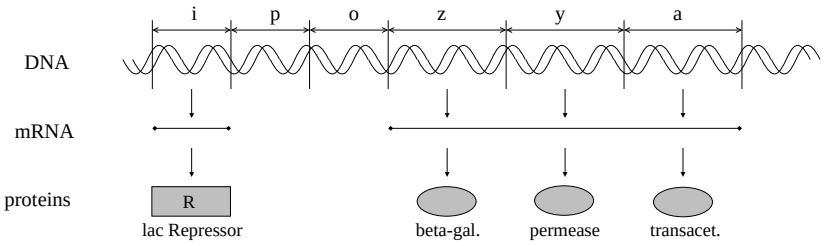

Fig. 2. The lactose operon.

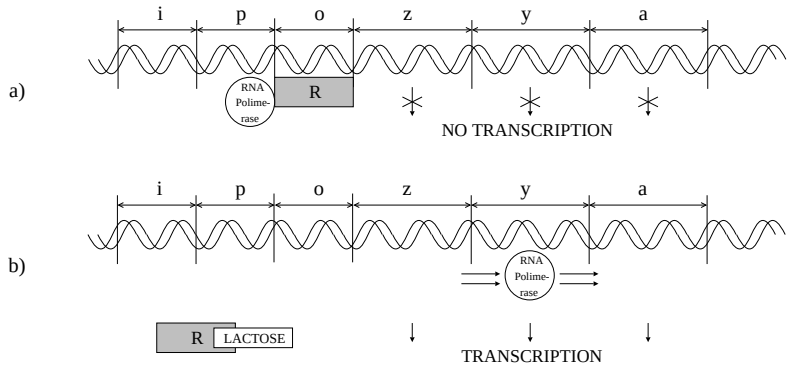

Fig. 3. The regulation process. In the absence of lactose (case a) the lac Repressor binds to gene o and precludes the RNA polymerase from transcribing genes z,y and a. When lactose is present (case b) it binds to and inactivates the lac Repressor.

3.1. Gene Regulation: the Lactose Operon in E.coli

E. coli is a bacterium often present in the intestine of many animals. As for most bacteria, it is often exposed to a constantly changing physical and chemical environment, and reacts to changes in its environment through changes in the kinds of proteins it produces.

In general, in order to save energy, bacteria do not synthesize degradative enzymes (which are proteins) unless the substrates for these enzymes are present in the environment. For example, E. coli does not synthesize the enzymes that degrade lactose (a sugar) unless lactose is in the environment. This phenomenon is called *enzyme induction* or, more generally, *gene regulation* since it is obtained by controlling the transcription of some genes into the corresponding proteins.

Let us consider the lactose degradation example in E. coli. Two enzymes are required to start the breaking process: (i) the *lactose permease*, which is incorporated in the membrane of the bacterium and actively transports the sugar into the cell (without this enzyme lactose can enter the bacterium anyway, but much more slowly); (ii) the *beta galactosidase*, which splits lactose into glucose and galactose. The bacterium produces also the *transacetylase* enzyme, whose role is marginal, but related with the usage of lactose.

The sequence of genes in the DNA of E. coli which produces the described enzymes, is known as the *lactose operon* (see Figure 2). It is composed by six genes: the first three (i, p, o) regulate the production of the enzymes, and the last three (z, y, a), called *structural genes*, are transcribed (when allowed) into the mRNA for beta galactosidase, lactose permease and transacetylase, respectively[1].

The regulation process is as follows (see Figure 3): gene i encodes the *lac Repressor*, which, in the absence of lactose, binds to gene o (the *operator*). Transcription of structural genes into mRNA is performed by the RNA polymerase enzyme, which usually binds to gene p (the *promoter*) and scans the operon from left to right by transcribing the three structural genes z, y and a into a single mRNA fragment. When the lac Repressor is bound to gene o, it becomes an obstacle for the RNA polymerase, and transcription of the

[1] We recall that in protein synthesis first the DNA of one or more genes is transcribed into a piece of mRNA, then the mRNA is translated into one or more proteins.

structural genes is not performed. On the other hand, when lactose is present inside the bacterium, it binds to the Repressor and this cannot stop any more the activity of the RNA polymerase. In this case transcription is performed and the three enzymes for lactose degradation are synthesized.

Now we describe how to model the gene regulation process with CLS. For the sake of simplicity we give a partial model, in the sense that we describe how the transcription of the structural genes is activated when the lactose is in the environment, but we do not describe how the transcription of such genes is stopped when the lactose disappears. Moreover, in order to simplify the example, we assume that genes are transcribed directly into proteins (thus avoiding the modelling of the mRNA), that the lac Repressor is transcribed from gene i without the need of the RNA polymerase and that it can be produced only once. Finally, we assume that one RNA polymerase is present inside the bacterium.

We model the membrane of the bacterium as the looping sequence $(m)^L$, where the symbol m generically denotes the whole membrane surface in normal conditions. Moreover, we model the lactose operon as the sequence $lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA$ ($lacI$–A for short), in which each element corresponds to a gene, and we replace $lacO$ with RO in the sequence when the lac Repressor is bound to gene o. When the lac Repressor is unbound, it is modelled by the symbol $repr$. Finally, we model the RNA polymerase as the symbol $polym$, a molecule of lactose as the symbol $LACT$, and beta galactose, lactose permease and transacetylase enzymes as symbols $betagal, perm$ and $transac$, respectively.

When no lactose is present, the bacterium is modelled by the following term:

$$Ecoli \ ::= \ (m)^L \rfloor (lacI \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA \mid polym)$$

The transcription of the DNA is modelled by the following set of rules:

$$\begin{aligned}
lacI \cdot \widetilde{x} &\mapsto lacI' \cdot \widetilde{x} \mid repr && \text{(R1)}\\
polym \mid \widetilde{x} \cdot lacP \cdot \widetilde{y} &\mapsto \widetilde{x} \cdot PP \cdot \widetilde{y} && \text{(R2)}\\
\widetilde{x} \cdot PP \cdot lacO \cdot \widetilde{y} &\mapsto \widetilde{x} \cdot lacP \cdot PO \cdot \widetilde{y} && \text{(R3)}\\
\widetilde{x} \cdot PO \cdot lacZ \cdot \widetilde{y} &\mapsto \widetilde{x} \cdot lacO \cdot PZ \cdot \widetilde{y} && \text{(R4)}\\
\widetilde{x} \cdot PZ \cdot lacY \cdot \widetilde{y} &\mapsto \widetilde{x} \cdot lacZ \cdot PY \cdot \widetilde{y} \mid betagal && \text{(R5)}\\
\widetilde{x} \cdot PY \cdot lacA &\mapsto \widetilde{x} \cdot lacY \cdot PA \mid perm && \text{(R6)}\\
\widetilde{x} \cdot PA &\mapsto \widetilde{x} \cdot lacA \mid transac \mid polym && \text{(R7)}
\end{aligned}$$

Rule (R1) describes the transcription of gene i into the lac Repressor. After transcription $lacI$ becomes $lacI'$ to avoid further productions of the lac Repressor. Rule (R2) describes the binding of the RNA polymerase to gene p. We denote the complex formed by the binding RNA polymerase to a gene $lac\_$ with the symbol $P\_$. Rules (R3)–(R6) describe the scanning of the DNA performed by the RNA polymerase and the consequent production of enzymes. Rule (R3) can be applied (and the scanning can be performed) only when the sequence contains $lacO$ instead of RO, that is when the lac Repressor is not bound to gene o. Finally, in rule (R7) the RNA polymerase terminates the scanning and releases the sequence.

The following rules describe the binding of the lac Repressor to gene o, and what happens when lactose is present in the environment of the bacterium:

$$\begin{aligned}
repr \mid \widetilde{x} \cdot lacO \cdot \widetilde{y} &\mapsto \widetilde{x} \cdot RO \cdot \widetilde{y} && \text{(R8)}\\
LACT \mid (m \cdot \widetilde{x})^L \rfloor X &\mapsto (m \cdot \widetilde{x})^L \rfloor (X \mid LACT) && \text{(R9)}\\
\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT &\mapsto \widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT && \text{(R10)}
\end{aligned}$$

Rule (R8) describes the binding of the lac Repressor to gene o, rule (R9) models the passage of the lactose through the membrane of the bacterium and rule (R10) the removal of the lac Repressor from gene o operated by the lactose. In this rule the symbol $RLACT$ denotes the binding of the lactose to the lac Repressor.

Finally, the following rules describe the behaviour of the enzymes synthesized when lactose is present, and their degradation:

$$\begin{aligned}
(\widetilde{x})^L \rfloor (perm \mid X) &\mapsto (perm \cdot \widetilde{x})^L \rfloor X && \text{(R11)}\\
LACT \mid (perm \cdot \widetilde{x})^L \rfloor X &\mapsto (perm \cdot \widetilde{x})^L \rfloor (LACT \mid X) && \text{(R12)}\\
betagal \mid LACT &\mapsto betagal \mid GLU \mid GAL && \text{(R13)}
\end{aligned}$$

$$perm \quad \mapsto \quad \epsilon \qquad \text{(R14)}$$
$$betagal \quad \mapsto \quad \epsilon \qquad \text{(R15)}$$
$$transac \quad \mapsto \quad \epsilon \qquad \text{(R16)}$$

Rule (R11) describes the incorporation of the lactose permease in the membrane of the bacterium, rule (R12) the transportation of lactose from the environment to the interior performed by the lactose permease, and rule (R13) the decomposition of the lactose into glucose (denoted GLU) and galactose (denoted GAL) performed by the beta galactosidase. Finally, rules (R14),(R15) and (R16) describe degradation of the lactose permease, the beta galactosidase and the transacetylase enzymes, respectively.

Let us denote the set of rewrite rules $\{(\text{R1}), \ldots, (\text{R16})\}$ as $\mathcal{R}_{lac}$, and the lactose operon $lacI' \cdot lacP \cdot lacO \cdot lacZ \cdot lacY \cdot lacA$, after the production of the lac Repressor, as $lacI'$–A. An example of possible sequence of transitions which can be performed by the term $Ecoli$ by applying rules in $\mathcal{R}_{lac}$ is the following:

$$\begin{aligned} Ecoli &\overset{\Box}{\Longrightarrow} \big(m\big)^L \rfloor (lacI' \cdot lacP \cdot RO \cdot lacZ \cdot lacY \cdot lacA \mid polym) \\ &\overset{LACT|\Box}{\Longrightarrow} \big(m\big)^L \rfloor (lacI'\text{–}A|polym|RLACT) \\ &\overset{\Box}{\Longrightarrow} \big(perm \cdot m\big)^L \rfloor (lacI'\text{–}A|betagal|transac|polym|RLACT) \\ &\overset{LACT|\Box}{\Longrightarrow} \big(perm \cdot m\big)^L \rfloor (lacI'\text{–}A|betagal|transac|polym|RLACT|GLU|GAL) \end{aligned}$$

In the example, by applying rules (R1) and (R8), $Ecoli$ produces the lac Repressor, which binds to gene o in the lactose operon. Then, the bacterium interacts with an environment containing a molecule of lactose (represented by the context $LACT|\Box$): by applying rule (R9) the lactose enters the membrane of the bacterium and by applying rule (R10) it binds to the lac Repressor. Then, a sequence of internal transitions are performed by applying rules (R2)–(R7) and (R11): the result is the transcription of the structural genes and the incorporation of the lactose permease in the membrane of the bacterium. Finally, the term interacts with an environment containing another molecule of lactose, which enters the bacterium and is decomposed into GLU and GAL. The rules applied in this phase are (R12) and (R13).

Note that, if one starts from $Ecoli$, every time (R12) can be applied, also (R9) can be applied giving the same results up to $\equiv$. Therefore, rule (R12) seems to be redundant. Nevertheless, rule (R12) describes a precise phenomenon, namely the action performed by the lactose permease, which is modelled by no other rule. The difference between rules (R9) and (R12) is that the latter describes a much faster event. However, since quantitative aspects are not considered in the calculus, the difference between the two rules does not appear.

Now we use the weak bisimulation defined on CLS systems to verify a simple property of the described system, namely that by starting from a situation in which the lac Repressor is bound to gene o, and none of the three enzymes produced by the lactose operon is present (which is a typical stable state of the system), production of such enzymes can start only if lactose appears.

In order to verify this property with the bisimulation relation we defined, we need to modify the rules of the model in such a way that the event of starting the production of the three enzymes becomes observable. We can obtain this result, for instance, by replacing rule (R10) with the rule

$$\big(\widetilde{w}\big)^L \rfloor (\widetilde{x} \cdot RO \cdot \widetilde{y} \mid LACT \mid X) \mid START \quad \mapsto \quad \big(\widetilde{w}\big)^L \rfloor (\widetilde{x} \cdot lacO \cdot \widetilde{y} \mid RLACT \mid X) \qquad \text{(R10bis)}$$

We choose to modify (R10) because we know that, after applying rule (R10), the three enzymes can be produced freely, and we add to the rule the interaction with the artificial element $START$ in the environment (namely outside the looping sequence $\big(\widetilde{w}\big)^L$ representing the membrane of the bacterium) in order to obtain $\Box|START$ as a transition label every time the rule is applied to the term. The property we want to verify is satisfied, for some ground terms T_1, T_2 and T_3, by the system $\langle T_1, \mathcal{R}\rangle$, where $\mathcal{R}$ consists of the following four rules:

$$T_1 \mid LACT \quad \mapsto \quad T_2 \qquad \text{(R1')} \qquad\qquad T_2 \mid START \quad \mapsto \quad T_3 \qquad \text{(R3')}$$
$$T_2 \mid LACT \quad \mapsto \quad T_2 \qquad \text{(R2')} \qquad\qquad T_3 \mid LACT \quad \mapsto \quad T_3 \qquad \text{(R4')}$$

It can be proved that the system $\langle T_1, \mathcal{R}\rangle$ is weakly bisimilar to the system $\langle EcoliRO, (\mathcal{R}_{lac} \setminus \{R_{10}\}) \cup$

$\{(R_{10bis})\}\rangle$, where:

$$EcoliRO = \big(m\big)^L \rfloor lacI' \cdot PP \cdot RO \cdot lacZ \cdot lacY \cdot lacA$$

In particular, the bisimulation relation associates (the system containing) term T_1 with (the system containing) term $EcoliRO$, term T_2 with all the terms representing a bacterium containing at least one molecule of lactose with the Lac repressor bound to gene o, and, finally, term T_3 with all the terms in which the repressor has left gene o and is bound to the lactose.

3.2. Signal Transduction: the EGF Signalling Pathway

In Biology, signal transduction refers to any process by which a cell converts one kind of signal or stimulus into another. Signals are typically proteins that may be present in the environment of the cell. In order to be able to receive the signal, namely to recognize that the corresponding protein is available in the environment, a cell exposes some receptors on its external membrane. A receptor is a transmembrane protein that can bind to a signal protein on its extracellular end. When such a binding is established, the intracellular end of the receptor undergoes a conformational change that enables interaction with other proteins inside the cell. This typically causes an ordered sequence of biochemical reactions inside the cell, usually called signalling pathway, that are carried out by enzymes and may produce different effects on the cell behaviour.

A complex signal transduction cascade, that modulates cell proliferation, survival, adhesion, migration and differentiation, is based on a family of receptors called epidermal growth factor receptors (EGFRs). While EGFR signalling is essential for many normal morphogenic processes, the aberrant activity of these receptors has been shown to play a fundamental role in proliferation of tumour cells [Yar01, YS01]. Epidermal growth factor receptors are located on the cell surface and they are activated by the binding with a specific ligand (epidermal growth factor, EGF) to form a EGFR (ligand-receptor) complex. Upon activation EGFR undergoes a transition from a monomeric form to an active dimeric one. EGFR dimerization stimulates its intracellular phosphorylation which activates signalling proteins. These activated signalling proteins (effector proteins) initiate several signal transduction cascades, leading to DNA synthesis and cell proliferation. After the activation of effector proteins, ligand-receptor dimers are internalized in endosomes. In a normal process an ubiquitine ligase, known as Cbl, is involved: Cbl binds an ubiquitin protein to the dimer (ubiquitination). The ubiquitin protein targets the dimers for lysosomal degradation (see Figure 4).

Many types of oncogenic viruses exploit the EGFR signalling cascade by manipulating its components. When the vCbl, the viral oncogenic form of Cbl, is present, EGFRs are recycled to the cellular membrane (instead of being sent to lysosomes for degradation), thus promoting cellular proliferation (see Figure 4a).

The human papilloma virus acts in a slightly different way. It produces an enzyme, E5, that appears to block the degradation of activated dimers by inhibiting the lysosomal ATPase (see Figure 4b). The effect is to return the internalized receptors to the cellular membrane where they may again bind ligands and stimulate proliferative pathways [SHJ+93].

Now we give the CLS model of the first steps of the signalling pathway, of the internalization and degradation of the ligand–receptor dimers, and of the activity of the vCbl and E5 viral enzymes. We model the cell membrane as a looping sequence $\big(m\big)^L$, where, as in the previous application, the symbol m generically denotes the whole membrane surface in normal conditions. The endosomal and lysosomal membranes are modelled as the looping sequences $\big(ENDO\big)^L$ and $\big(LYSO\big)^L$, respectively. Moreover, we model the signal and receptor proteins as the elements EGF and $EGFR$, respectively. A signal–receptor complex is denoted as $CMPLX$ and a dimer composed by two of such complexes is denoted either as DIM, before phosphorylation, or as $DIMp$, after phosphorylation, or as $DIMpUB$, after phosphorylation and ubiquitination. We denote an effector protein with EFF, that becomes $EFFp$ after phosphorylation. Finally, we denote the Cbl, vCbl, E5 and ATPase enzymes and the binding between E5 and ATPase as symbols $CBL, vCBL, E5, ATPase$ and $ATPaseE5$, respectively.

The first steps of the EGF signalling pathway are described by the following rules:

$$EGF \mid \big(EGFR \cdot \widetilde{x}\big)^L \rfloor X \quad \mapsto \quad \big(CMPLX \cdot \widetilde{x}\big)^L \rfloor X \tag{R1}$$

$$\big(CMPLX \cdot \widetilde{x} \cdot CMPLX \cdot \widetilde{y}\big)^L \rfloor X \quad \mapsto \quad \big(DIM \cdot \widetilde{x} \cdot \widetilde{y}\big)^L \rfloor X \tag{R2}$$

$$\big(DIM \cdot \widetilde{x}\big)^L \rfloor X \quad \mapsto \quad \big(DIMp \cdot \widetilde{x}\big)^L \rfloor X \tag{R3}$$

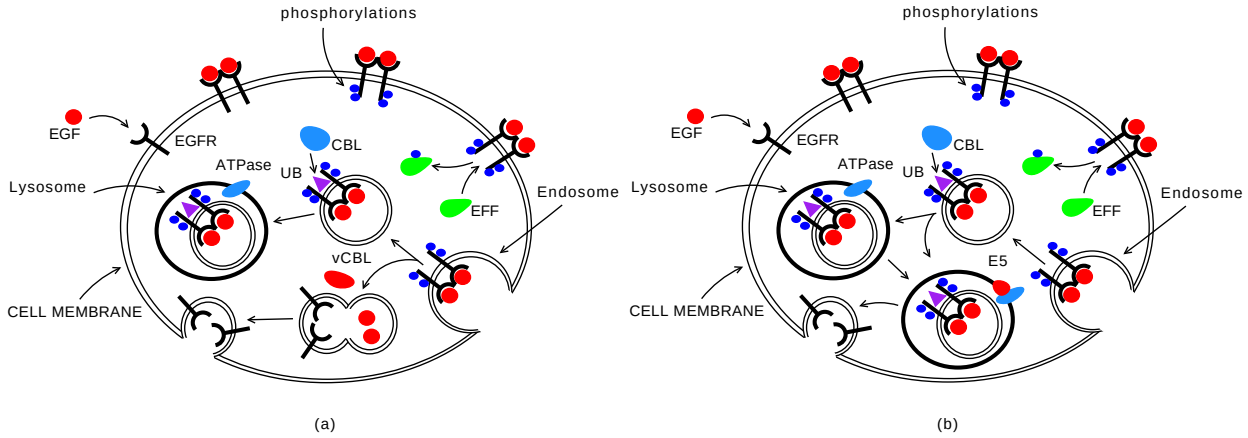

Fig. 4. Degradation and recycling of receptors in the EGF pathway. After ubiquitination, dimers are sent to the Lysosome for degradation. The vCbl viral enzyme (a) prevents ubiquitination, whereas the E5 viral enzyme (b) prevents degradation inside the Lysosome. Both the activities of the two viral enzymes favor recycling of the receptors.

$$\left(DIMp \cdot \widetilde{x}\right)^L \rfloor (X \mid EFF) \quad \mapsto \quad \left(DIMp \cdot \widetilde{x}\right)^L \rfloor (X \mid EFFp) \tag{R4}$$

Rule (R1) describes the binding of the EGF signal protein with a receptor EGFR on the cell membrane. The result of the binding is a signal–receptor complex whose dimerization is described by rule (R2). Rule (R3) describes the phosphorylation (activation) of a dimer, which enables the propagation of the signal inside the cell by means of phosphorylation of effector proteins (rule (R4)).

The following rules describe internalization and degradation of signal–receptor complexes:

$$\left(DIMp \cdot \widetilde{x}\right)^L \rfloor X \quad \mapsto \quad \left(\widetilde{x}\right)^L \rfloor (X \mid \left(ENDO \cdot DIMp\right)^L \rfloor \epsilon) \tag{R5}$$

$$CBL \mid \left(ENDO \cdot DIMp\right)^L \rfloor X \quad \mapsto \quad CBL \mid \left(ENDO \cdot DIMpUB\right)^L \rfloor X \tag{R6}$$

$$\left(ENDO \cdot DIMpUB\right)^L \rfloor X \mid \left(LYSO \cdot \widetilde{x}\right)^L \rfloor Y \quad \mapsto \quad \left(LYSO \cdot \widetilde{x}\right)^L \rfloor (Y \mid \left(ENDO \cdot DIMpUB\right)^L \rfloor X) \tag{R7}$$

$$\left(LYSO \cdot ATPase\right)^L \rfloor X \quad \mapsto \quad \left(LYSO \cdot ATPase\right)^L \rfloor \epsilon \tag{R8}$$

Rule (R5) describes the internalization of an active dimer in an endosome. Rule (R6) describes the ubiquitination of the internalized dimer perfomed by the Cbl enzyme, that allows the lysosome to engulf the whole endosome (rule (R7)). Finally, rule (R8) describes the degradation of the lysosome content caused by the activity of the ATPase enzyme.

The following last four rules describe the activity of the vCbl and E5 viral enzymes:

$$vCBL \mid \left(ENDO \cdot DIMp\right)^L \rfloor X \quad \mapsto \quad vCBL \mid \left(ENDO \cdot EGFR \cdot EGFR\right)^L \rfloor (X \mid EGF \mid EGF) \tag{R9}$$

$$\left(\widetilde{x}\right)^L \rfloor (X \mid \left(ENDO \cdot EGFR \cdot EGFR\right)^L \rfloor Y) \quad \mapsto \quad \left(EGFR \cdot EGFR \cdot \widetilde{x}\right)^L \rfloor (X \mid \left(ENDO\right)^L \rfloor Y) \tag{R10}$$

$$E5 \mid \left(LYSO \cdot ATPase\right)^L \rfloor X \quad \mapsto \quad \left(LYSO \cdot ATPaseE5\right)^L \rfloor X \tag{R11}$$

$$\left(LYSO \cdot ATPaseE5\right)^L \rfloor (X \mid \left(ENDO \cdot DIMpUB\right)^L \rfloor Y) \quad \mapsto$$
$$\left(LYSO \cdot ATPaseE5\right)^L \rfloor X \mid \left(ENDO \cdot EGFR \cdot EGFR\right)^L \rfloor (EGF \mid EGF) \tag{R12}$$

Rule (R9) describes the activity of the vCbl enzyme, that decomposes an active dimer into its signal and receptor components. Rule (R10) describes the recycling of the receptors after vCbl intervention. Rule (R11) describes the inhibition of the lysosomal ATPase enzyme performed by E5. Finally, rule (R12) describes the decomposition of a dimer into its signal and receptor components that occurs in the lysosome when the ATPase enzyme is inhibited. Also in this case the receptors can be recycled by applying rule (R10).

Now we can use bisimulations to study some properties of the described system. In particular, we can study the effect of the viral infections on the reaction of the cell to external stimuli.

Let us consider a cell with two receptors on its external membrane, namely the term

$$CELL ::= \left(m \cdot EGFR \cdot EGFR\right)^L \rfloor (CBL \mid \left(LYSO \cdot ATPase\right)^L \rfloor \epsilon)$$

and let us denote the same cell when it is infected by a virus and contains either a vCbl or an E5 enzyme as

$$CELL\_vCBL ::= \left(m \cdot EGFR \cdot EGFR\right)^L \rfloor (CBL \mid vCBL \mid \left(LYSO \cdot ATPase\right)^L \rfloor \epsilon)$$

and

$$CELL\_E5 ::= \left(m \cdot EGFR \cdot EGFR\right)^L \rfloor (CBL \mid E5 \mid \left(LYSO \cdot ATPase\right)^L \rfloor \epsilon)$$

respectively. Moreover, let us denote the set of rules $\{(R1), \ldots, (R12)\}$ as $\mathcal{R}_{egf}$.

It is easy to prove that $\langle CELL, \mathcal{R}_{egf}\rangle \approx \langle T_1, \mathcal{R}\rangle$, for some term T_1 and where $\mathcal{R}$ consists of the following rules:

$$T_1 \mid EGF \quad \mapsto \quad T_2 \qquad \text{(R1')} \qquad\qquad T_2 \mid EGF \quad \mapsto \quad T_3 \qquad \text{(R2')}$$

for some terms T_2 and T_3. On the other side, it holds that $\langle CELL\_vCBL, \mathcal{R}_{egf}\rangle \approx \langle T_a, \mathcal{R}'\rangle$ for some term T_a and where $\mathcal{R}$ consists of the following rules:

$$T_a \mid EGF \quad \mapsto \quad T_b \qquad \text{(R1")} \qquad\qquad T_c \quad \mapsto \quad T_d \qquad \text{(R3")}$$
$$T_b \mid EGF \quad \mapsto \quad T_c \qquad \text{(R2")} \qquad\qquad T_c \quad \mapsto \quad T_a \qquad \text{(R4")}$$

for some terms T_b, T_c and T_d. Since $\langle T_1, \mathcal{R}\rangle \not\approx \langle T_a, \mathcal{R}'\rangle$, as the first can perform only a sequence of two $\xrightarrow{\square \mid EGF}$ transitions and the second possibly an infinite sequence of such transitions, it holds that $\langle CELL, \mathcal{R}_{egf}\rangle \not\approx \langle CELL\_vCBL, \mathcal{R}_{egf}\rangle$, namely that the infection of the cell by a virus producing the vCbl enzyme causes an observable change in the cell behaviour. Since the set of rewrite rules in the two systems is the same, it holds that $CELL \not\approx CELL\_vCBL$ as well.

It is easy to see that also $\langle CELL\_E5, \mathcal{R}_{egf}\rangle \approx \langle T_a, \mathcal{R}'\rangle$ holds, namely that also the production of the viral enzyme E5 causes an observable change in the cell behaviour. More precisely, since both $\langle CELL\_vCBL, \mathcal{R}_{egf}\rangle$ and $\langle CELL\_E5, \mathcal{R}_{egf}\rangle$ are weakly bisimilar to $\langle T_a, \mathcal{R}'\rangle$, we have $\langle CELL\_vCBL, \mathcal{R}_{egf}\rangle \approx \langle CELL\_E5, \mathcal{R}_{egf}\rangle$. This means that, even if the two enzymes interact with the internal machinery of the cell in different ways, the effect on the global cell behaviour is the same. Moreover, since the set of rewrite rules in the two systems is the same, it holds also that $CELL\_vCBL \approx CELL\_E5$, and this is a stronger proof of the equivalence of the effects of the two viruses because, by the congruence property of $\approx$, it holds that the behaviour of the two infected cells will be the same in any context (for instance in the presence and in the absence of EGF signals or when there are other cells in the environment).

4. Brane Calculi

Brane Calculi are a family of process calculi specialized in the description of membrane activity. We consider the simplest calculus of the family, the PEP calculus. In [BMM+06a, Mil07] we gave an operationally adequate encoding of the PEP calculus into CLS. In this section we recall the definition of the PEP calculus, then we define a labelled semantics for the PEP calculus and bisimulation relations. We have chosen the PEP calculus for its simplicity. We could give labelled semantics and bisimulations for other calculi of the family by following the same approach.

4.1. The PEP Calculus

The syntax and the semantics of the PEP calculus is summarized in Figure 5. Terms are systems. Systems consist of composition of systems, $\circ$, with unit $\diamond$. Replication ! is used to model the notion of "multitude" of systems. Systems can be membrane containing systems, $\sigma(\!|P|\!)$. Membranes can be parallel composition $\sigma|\sigma'$ with unit $\mathbf{0}$, or replication of membranes, or action prefixing.

Actions are: *phagocytosis*, denoted ϕ_n, incorporates one external membrane into another by "engulfing" it; *exocytosis*, denoted by ε_n, is the reverse process; *pinocytosis*, denoted by $\circledcirc$, engulfs zero external membranes. Phagocytosis and exocytosis have co-actions that are intended to interact with, indicated by the symbol $^\perp$. Pinocytosis does not have a co-action. Figure 6 gives a pictorial representation of the three actions.

Syntax

$P, Q, R \ldots ::= \diamond \mid P \circ P \mid !P \mid \sigma(\!|P|\!)$ Systems

$\sigma, \tau, \rho, \ldots ::= \mathbf{0} \mid \sigma|\sigma \mid !\sigma \mid a.\sigma$ Branes

$a, b, c, \ldots ::= \phi_n \mid \phi_n^{\perp}(\sigma) \mid \varepsilon_n \mid \varepsilon_n^{\perp} \mid \circledcirc(\sigma)$ Actions

Structural Congruence

The least congruence relation $\equiv$ satisfying the following axioms

$P \circ Q \equiv Q \circ P \qquad P \circ (Q \circ R) \equiv (P \circ Q) \circ R \qquad P \circ \diamond \equiv P$

$!\diamond \equiv \diamond \qquad !!P \equiv !P \qquad !P \equiv P \circ !P \qquad \mathbf{0}(\!|\diamond|\!) \equiv \diamond$

$\sigma|\tau \equiv \tau|\sigma \qquad \sigma|(\tau|\rho) \equiv (\sigma|\tau)|\rho \qquad \sigma|\mathbf{0} \equiv \sigma \qquad !\mathbf{0} \equiv \mathbf{0} \qquad !!\sigma \equiv !\sigma \qquad !\sigma \equiv \sigma|!\sigma$

Reduction Semantics

The least relation containing the following axioms, closed wrt $\_ \circ P$, $\sigma(\!|\_|\!)$ and $\equiv$

(phago) $\phi_n.\sigma|\sigma_0(\!|P|\!) \circ \phi_n^{\perp}(\rho).\tau|\tau_0(\!|Q|\!) \rightarrow \tau|\tau_0(\!|\rho(\!|\sigma|\sigma_0(\!|P|\!)|\!) \circ Q|\!)$

(exo) $\varepsilon_n^{\perp}.\tau|\tau_0(\!|\varepsilon_n.\sigma|\sigma_0(\!|P|\!) \circ Q|\!) \rightarrow P \circ \sigma|\sigma_0|\tau|\tau_0(\!|Q|\!)$

(pino) $\circledcirc(\rho).\sigma|\sigma_0(\!|P|\!) \rightarrow \sigma|\sigma_0(\!|\rho(\!|\diamond|\!) \circ P|\!)$

Fig. 5. The phago/exo/pino (PEP) calculus: syntax and semantics.

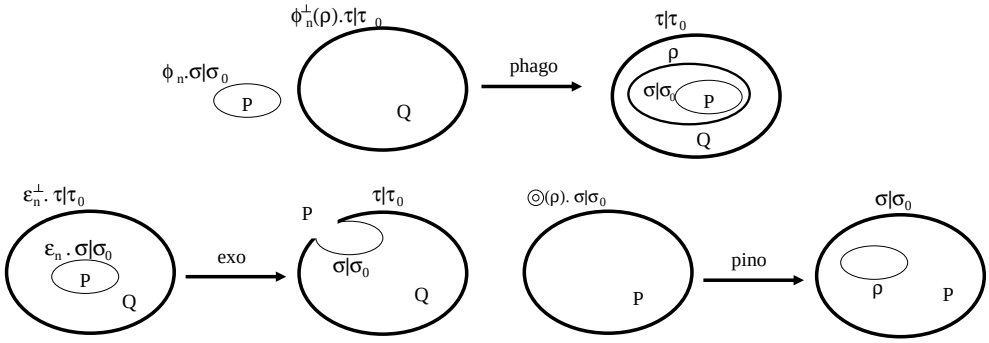

Fig. 6. Pictorial representation of phagocytosis, exocytosis and pinocytosis.

We consider a structural congruence relation $\equiv$ that describes associativity, commutativity, replication and unit elements of operators on systems and membranes. We denote with PEP the infinite set of Systems, and with $Branes$ the infinite set of membranes in the PEP calculus. Moreover, we denote with $\mathcal{N}$ the (possibly infinite) set of names n used as subscripts of Actions.

4.2. A Labelled Semantics for the PEP Calculus

In process calculi theory, a labelled semantics usually allows description of the potential behaviour of a process in terms of possible interactions with other processes that could occur in its environment. This is obtained by allowing the process performing as many transitions as are its active actions, each transition having the corresponding action as label and leading to a new process which corresponds to the result of the execution of the action. Moreover, labelled semantics include silent transitions, often labelled with τ, describing internal activity, namely interactions occurring between internal components of the process. Silent transitions are used also to describe actions that are performed by a single component of the process, without any interaction. Furthermore, if actions of the process calculus require parameters (for instance an action of

sending or receiving a message may require the transmitted message as parameter) then also the parameter is shown in the transition label.

In the PEP calculus the actions are phagocytosis, exocytosis and pinocytosis. The two former describe the interaction between two different membranes, while the latter is performed by a single membrane (causing a silent transition). We use $\_$ as the label for silent transitions.

Phagocytosis and exocytosis can be seen as communications: a membrane which is engulfed by another one can be seen as a membrane sending itself to the other, and a membrane which is expelled by another one can be seen as a membrane sending itself to the external environment. Hence, from a process calculi point of view, the message which is transmitted is the continuation of the process which performs the action of sending. As a consequence, for transitions corresponding to ϕ_n and ε_n actions we use as labels pairs $(\phi_n, \sigma(\!|P|\!))$ and $(\varepsilon_n, \sigma(\!|P|\!))$, respectively, in which $\sigma(\!|P|\!)$ is the continuation of the membrane performing ϕ_n and ε_n, respectively.

Now, we have to consider the $\phi_n^\perp$ and the $\varepsilon_n^\perp$ actions, namely the actions of receiving a message. In the first case, the case of phagocytosis, we have that the two membranes performing ϕ_n and $\phi_n^\perp$ are composed by $\circ$, hence each one is in the context of the other, and we can use $(\phi_n^\perp, \sigma(\!|P|\!))$ as label for the $\phi_n^\perp$ when the received message is $\sigma(\!|P|\!)$. In the case of exocytosis, instead, the process performing the ε_n action is not in the context of the one performing $\varepsilon_n^\perp$, but it is inside the membrane performing $\varepsilon_n^\perp$. Hence, the use of $(\varepsilon_n^\perp, \sigma(\!|P|\!))$ as a label for this action would be meaningless, as $\varepsilon_n^\perp$ does not cause a potential interaction with the environment, but a potential internal interaction. After this discussion, we conclude that the set of labels of the labelled semantics of the PEP calculus is $\mathcal{L} = \{(\phi_n, \sigma(\!|R|\!)), (\phi_n^\perp, \sigma(\!|R|\!)), (\varepsilon_n, \sigma(\!|R|\!)), \_ \mid n \in \mathcal{N}, R \in PEP, \sigma \in Branes\}$. Let ℓ range over this set.

Now, a system P that is able to perform a $(\phi_n, \sigma(\!|R|\!))$ transition, $P \xrightarrow{\phi_n, \sigma(\!|R|\!)} P'$, is a system which has a component $\sigma(\!|R|\!)$ that can enter a membrane, while a system Q that is able to perform a $(\phi_n^\perp, \sigma(\!|R|\!))$ transition, $Q \xrightarrow{\phi_n^\perp, \sigma(\!|R|\!)} Q'$, is a system that can engulf another system $\sigma(\!|R|\!)$. When P and Q are composed by $\circ$, they can evolve together, with a silent action, to a new system in which $\sigma(\!|R|\!)$ is inside Q'.

Definition 15 (Labelled Semantics). The labelled semantics of the PEP calculus is given by the labelled transition system generated by the inference rules in Figure 7. Terms in the rules are considered modulo structural congruence.

Rules (Ph1), (Ph2) and (Ph3) describe the behaviour of systems which can perform phagocytosis. Rule (Ph1) describes the evolution of a system which can be engulfed. Rule (Ph2) describes the evolution of a system which can engulf any other system $\sigma(\!|R|\!)$. Rule (Ph3) describes an actual phagocytosis which involves two systems.

Rules (E1) and (E2) describe the exocytosis process. Rule (E1) describes the behaviour of a system which can exit a membrane. Recall that, in the labelled semantics, an action represents the potentiality of a system when inserted in a suitable context, thus the transition $P \xrightarrow{\varepsilon_n, \sigma(\!|R|\!)} P'$, intuitively means that P has a component $\sigma(\!|R|\!)$ that, when inserted in a membrane τ, can abandon its membrane σ and also can get away from τ becoming R. For this reason, there is no corresponding transition with $\varepsilon_n^\perp$ as label. The possibility of a system to allow an internal system to get away does not depend on the context but on the internal state of the system. In fact, rule (E2) states that a system can allow an internal system to exit by a silent action, while membranes σ, τ and τ_0 coalesce (see also Figure 6).

Rule (Pi1) describes the pinocytosis process. Rules (Par1) and (Par2) state that an action of a system P can be observed also when P is composed by $\circ$ with other systems. Finally, rule (Br1) states that actions internal to a membrane cannot be observed from outside, and only the silent action is allowed in such a context.

4.3. Bisimulation Relations

We define strong and weak bisimulation relations on the labelled semantics of the PEP calculus. Usually, (strongly) bisimilar processes must be step by step able to perform transitions with the same labels. In the labelled semantics we have defined for the PEP calculus, labels are not simple objects: they may contain branes which can be arbitrarily complex. These branes that may occur in transition labels will become active parts of the considered PEP system once the transitions having them as labels have been performed.

$$\phi_n.\sigma|\sigma_0(\!|P|\!) \xrightarrow{\phi_n,\sigma|\sigma_0(\!|P|\!)} \diamond \quad \text{(Ph1)}$$

$$\phi_n^{\perp}(\rho).\tau|\tau_0(\!|Q|\!) \xrightarrow{\phi_n^{\perp},\sigma(\!|R|\!)} \tau|\tau_0(\!|\rho(\!|\sigma(\!|R|\!)|\!) \circ Q|\!) \quad \text{(Ph2)}$$

$$\varepsilon_n.\sigma|\sigma_0(\!|P|\!) \xrightarrow{\varepsilon_n,\sigma|\sigma_0(\!|P|\!)} \diamond \quad \text{(E1)}$$

$$\circledcirc(\rho).\sigma|\sigma_0(\!|P|\!) \dashrightarrow \sigma|\sigma_0(\!|P \circ \rho(\!|\diamond|\!)|\!) \quad \text{(Pi1)}$$

$$\frac{P \xrightarrow{\phi_n,\sigma(\!|R|\!)} P' \qquad Q \xrightarrow{\phi_n^{\perp},\sigma(\!|R|\!)} Q'}{P \circ Q \dashrightarrow P' \circ Q'} \quad \text{(Ph3)}$$

$$\frac{P \xrightarrow{\ell} P'}{P \circ Q \xrightarrow{\ell} P' \circ Q} \qquad \frac{Q \xrightarrow{\ell} Q'}{P \circ Q \xrightarrow{\ell} P \circ Q'} \quad \text{(Par1,Par2)}$$

$$\frac{P \xrightarrow{\varepsilon_n,\sigma(\!|R|\!)} P'}{\varepsilon_n^{\perp}.\tau|\tau_0(\!|P \circ Q|\!) \dashrightarrow R \circ \sigma|\tau|\tau_0(\!|Q \circ P'|\!)} \quad \text{(E2)}$$

$$\frac{P \dashrightarrow P'}{\sigma(\!|P|\!) \dashrightarrow \sigma(\!|P'|\!)} \quad \text{(Br1)}$$

Fig. 7. The phago/exo/pino (PEP) calculus: inference rules for the Labelled Semantics.

In the definition of the bisimulation relation we have to take into account the role of the brane used as a transition label. We could require that the transitions of two bisimilar systems are exactly the same, but this would be a too strong requirement. Instead, we require that the branes appearing in the transitions of two bisimilar systems are bisimilar too. In such a way we can ensure that when these branes will be activated, they will have the same behaviour, even if they are not syntactically identical.

The strong bisimulation relation for PEP systems is defined as follows.

Definition 16 (PEP Strong Bisimulation). A binary relation κ on PEP terms is a *strong bisimulation* if, given P and Q such that $P\kappa Q$, the following conditions hold:

$P \dashrightarrow P' \implies \exists Q'$ such that $Q \dashrightarrow Q'$ and $P'\kappa Q'$

$Q \dashrightarrow Q' \implies \exists P'$ such that $P \dashrightarrow P'$ and $Q'\kappa P'$

$P \xrightarrow{a,R} P' \implies \exists Q'$ such that $Q \xrightarrow{a,R'} Q', R\kappa R'$ and $P'\kappa Q'$

$Q \xrightarrow{a,R} Q' \implies \exists P'$ such that $P \xrightarrow{a,R'} P', R\kappa R'$ and $Q'\kappa P'$

The *strong bisimilarity* $\backsimeq$ is the largest of such relations.

As usual, the weak bisimulation relation for PEP systems differs from the strong relation as it allows systems to differ in the silent transitions they perform.

We denote the reflexive transitive closure of $\dashrightarrow$ as $\Longrightarrow$, namely $P \Longrightarrow P'$ if either $P \equiv P'$ or there exist $P_1, \ldots, P_n$ such that $P \dashrightarrow P_1 \dashrightarrow \ldots \dashrightarrow P_n \dashrightarrow P'$. We denote with $\stackrel{\ell}{\Longrightarrow}$ with $\ell \neq \_$ a composition of transitions $\Longrightarrow\stackrel{\ell}{\longrightarrow}\Longrightarrow$, namely $P \stackrel{\ell}{\Longrightarrow} P'$ if there exist P_1, P_2 such that $P \Longrightarrow P_1 \xrightarrow{\ell} P_2 \Longrightarrow P'$.

Definition 17 (PEP Weak Bisimulation). A binary relation κ on PEP terms is a *weak bisimulation* if, given P and Q such that $P\kappa Q$, the following conditions hold:

$P \dashrightarrow P' \implies \exists Q'$ such that $Q \Longrightarrow Q'$ and $P'\kappa Q'$

$Q \dashrightarrow Q' \implies \exists P'$ such that $P \Longrightarrow P'$ and $Q'\kappa P'$

$P \xrightarrow{a,R} P' \implies \exists Q'$ such that $Q \stackrel{a,R'}{\Longrightarrow} Q', R\kappa R'$ and $P'\kappa Q'$

$Q \xrightarrow{a,R} Q' \implies \exists P'$ such that $P \stackrel{a,R'}{\Longrightarrow} P', R\kappa R'$ and $Q'\kappa P'$

The *weak bisimilarity* $\approxeq$ is the largest of such relations.

As usual, strong bisimilarity between systems implies weak bisimilarity, namely the former relation is contained in the latter.

Lemma 5. Given $P, Q \in PEP$, it holds that $P \simeq Q \implies P \approx Q$

Proof. Follows directly from the definitions of the two relations. □

It holds that both bisimilarities are congruences. This, in process calculi theory, means that given two bisimilar systems P and Q, for any context C in which P and Q could be placed, it holds that $C[P]$ and $C[Q]$ are still bisimilar systems. Contexts for the PEP calculus can be defined as follows.

Definition 18 (Contexts). *Contexts* of PEP systems are given by the following grammar:

$$C ::= \Box \quad | \quad C \circ R \quad | \quad R \circ C \quad | \quad !C \quad | \quad \sigma(\!|C|\!)$$

where R is a PEP system and $\Box$ denotes the empty context.

As usual, $C[P]$ denotes the application of the context C to the PEP system P, that is a new PEP system obtained by replacing $\Box$ with P in the context C, and $C[C']$ denotes context composition, that is a new context obtained by replacing $\Box$ with C' in the context C.

The congruence results are stated in the following two theorems.

Theorem 5 (Strong Congruence). The strong bisimilarity on PEP systems $\simeq$ is a congruence.

Proof. The proof that $\simeq$ is an equivalence relation is trivial. In order to prove that it is also a congruence we have to prove that given $P, Q \in PEP$ such that $P \simeq Q$, it holds that $C[P] \simeq C[Q]$, namely that:

(i) for any transition $C[P] \stackrel{-}{\rightarrow} P'$ there exists a corresponding transition $C[Q] \stackrel{-}{\rightarrow} Q'$ such that $P' \simeq Q'$,

(ii) for any transition $C[P] \xrightarrow{a,R} P'$ there exists a corresponding transition $C[Q] \xrightarrow{a,R'} Q'$ such that $P' \simeq Q'$ and $R \simeq R'$.

We can prove this by induction on the structure of C, and, for each possible case, by induction on the derivation of the performed transitions.

- The *base case* is $C = \Box$. This case is trivial as $C[P] = P$ and $C[Q] = Q$.
- *Induction step*
  - If $C \equiv R \circ C'$ and $C \equiv C' \circ R$ we have that $C[P] \equiv R \circ C'[P]$ and $C[Q] \equiv R \circ C'[Q]$. A transition $C[P] \xrightarrow{\ell} P'$ can be performed either by R in isolation, or by $C'[P]$ in isolation, or through an interaction between the two components. The first case is trivial as R can perform the same transitions also when it occurs in $C[Q]$. In the second case we should just apply the induction hypothesis. In the third case we have that ℓ is equal to $\_$, and R and $C'[P]$ are able to perform two transitions representing phagocytosis. By induction hypothesis we have that $C'[Q]$ can perform an equivalent transition and interact with R in $C[Q]$.
  - If $C = !C'$, by the definition of the structural congruence relation we have $!C'[P] \equiv !C'[P] \circ C'[P] \circ C'[P]$, and $!C'[Q] \equiv !C'[Q] \circ C'[Q] \circ C'[Q]$. If $!C'[P] \stackrel{-}{\rightarrow} P'$, then there exist P'' such that $C'[P] \circ C'[P] \stackrel{-}{\rightarrow} P''$ and $P' \equiv P'' \circ !C'[P]$. By induction hypothesis, there exists Q'' such that $C'[Q] \circ C'[Q] \stackrel{-}{\rightarrow} Q''$ with $P'' \simeq Q''$. Similarly, $!C'[P] \equiv !C'[P] \circ C'[P]$ and $!C'[Q] \equiv !C'[Q] \circ C'[Q]$. If $!C'[P] \xrightarrow{a,R} P'$, then there exist P'' such that $C'[P] \xrightarrow{a,R'} P''$ and $P' \equiv P'' \circ !C'[P]$. By induction hypothesis, there exists Q'' such that $C'[Q] \xrightarrow{a,R'} Q''$ with $P'' \simeq Q''$ and $R \simeq R'$. Hence, for any transition performed by $!C[P]$ there exists an equivalent transition performed by $!C[P]$ leading to bisimilar systems.
  - If $C = \sigma(\!|C'|\!)$ the induction hypothesis can be applied trivially. □

Theorem 6 (Weak Congruence). The weak bisimilarity on PEP systems $\approx$ is a congruence.

Proof. The proof is similar to the proof of Theorem 5. □

As an example of weakly bisimilar PEP systems let us consider the systems $P = \phi^{\perp}_{n1}(\varepsilon^{\perp}_{n2}).\varepsilon^{\perp}_{n3}(\!|\diamond|\!)$ and $Q = \phi^{\perp}_{n1}(\phi^{\perp}_{n4}.\varepsilon^{\perp}_{n2}).\varepsilon^{\perp}_{n3} \circledcirc (\phi_{n4})(\!|\diamond|\!)$. System P is a membrane that can engulf another one present in the environment by a phagocytosis (indexed $n1$). This has as an effect that the system contains the

engulfed membrane surrounded by an intermediate membrane. An action of exocytosis (indexed $n2$) can be performed by this membrane, while another action of exocytosis (indexed $n3$) can be performed by the outer membrane. The engulfed membrane may interact with the mentioned actions of exocytosis with its ejection in the environment as a possible result. System Q, similarly to system P, can engulf a membrane present in the environment by a phagocytosis (indexed $n1$). Now, the intermediate membrane, before performing the exocytosis (indexed $n2$), must perform a phagocytosis (indexed $n4$). This can be done by interacting with the membrane created by the action of pinocytosis performed by the outer membrane. As for system P, the engulfed membrane may be ejected in the environment.

We have that the only transition that can be performed by P is

$$P \xrightarrow{\phi_{n1}^{\perp}, P'} \varepsilon_{n3}^{\perp}(\!|\varepsilon_{n2}^{\perp}(\!|P'|\!)|\!)$$

for some P'. System Q can mimic this transition by performing the following sequence of transitions

$$Q \xrightarrow{\phi_{n1}^{\perp}, Q'} \varepsilon_{n3}^{\perp}|\circledcirc(\phi_{n4})(\!|\phi_{n4}^{\perp}.\varepsilon_{n2}^{\perp}(\!|Q'|\!)|\!) \xrightarrow{-} \varepsilon_{n3}^{\perp}(\!|\phi_{n4}(\!|\diamond|\!)\circ\phi_{n4}^{\perp}.\varepsilon_{n2}^{\perp}(\!|Q'|\!)|\!) \xrightarrow{-} \varepsilon_{n3}^{\perp}(\!|\varepsilon_{n2}^{\perp}(\!|\mathbf{0}(\!|\diamond|\!)\circ Q'|\!)|\!) \equiv \varepsilon_{n3}^{\perp}(\!|\varepsilon_{n2}^{\perp}(\!|Q'|\!)|\!)$$

for some Q' weakly bisimilar to P'. This is not the only sequence of transitions performed by Q that can mimic the transition performed by P. Another one can be obtained by swapping the order of execution of the first two steps, namely by executing immediately the $\circledcirc$ action and then the $\phi_{n1}^{\perp}$ action. Moreover, other transitions (with label $\_$) could be performed by executing some actions of P' and Q' after their phagocytosis. However, since $P' \simeq Q'$, any of these transitions of one of the two systems can be simulated by the other system. Finally, since weak bisimulation is a congruence and since $P' \simeq Q'$, we have that the state reached after the transition performed by P, namely $\varepsilon_{n3}^{\perp}(\!|\varepsilon_{n2}^{\perp}(\!|P'|\!)|\!)$, is weakly bisimilar to the state reached after the sequence of transitions performed by Q, namely $\varepsilon_{n3}^{\perp}(\!|\varepsilon_{n2}^{\perp}(\!|Q'|\!)|\!)$, and hence P and Q are weakly bisimilar.

5. Comparing PEP and CLS Bisimilarities

In this section we briefly recall the encoding we gave in [BMM+06a] for translating PEP systems into CLS terms. We are then able to compare bisimulations defined for the PEP calculus with the bisimulations defined for CLS. Namely, we show that the translations of PEP systems which are bisimilar result in bisimilar CLS terms.

The motivation for comparing the bisimulations of CLS and PEP is twofold. First, the fact that bisimilarity is preserved confirms the correctness of the encoding. In fact, it means that the encoding can be used even when the finer semantics for CLS and PEP defined in this paper are considered instead of the original ones. Second, it shows that the definition of labelled semantics and bisimulations in two different formalisms for the same biological systems and with two different approaches has lead to comparable notions of equivalences. This means that the notion of behavioral equivalence is proper and meaningful for biological systems and does not depend on the language used to describe them.

5.1. Encoding of the PEP Calculus into CLS

The encoding of a system of the PEP calculus into a CLS term results in a pair composed of a CLS sequence and a set of alphabet symbols.

Operators and actions of the encoded system are translated into elements of the sequence. More precisely, $\diamond$ is translated into $\mathbf{0}$, and the three operators on systems $\_\circ\_$, $!\_$ and $\_(\!|\_|\!)$ are translated into *circ*,*bangS* and *brane*, respectively, with $\mathbf{0}, circ, bangS, brane \in \mathcal{E}$. Moreover, as regards branes, $\mathbf{0}$ is translated into $\mathbf{0}$, $\_|\_$ into *par* and $!\_$ into *bangB*, with $par, bangB \in \mathcal{E}$. Phagocytosis and exocytosis actions are translated into sequences of two elements, namely $\phi_n, \phi_n^{\perp}, \varepsilon_n$ and $\varepsilon_n^{\perp}$ are translated into $\phi \cdot n, \phi^{\perp} \cdot n, \varepsilon \cdot n$ and $\varepsilon^{\perp} \cdot n$, respectively. Finally, pinocytosis $\circledcirc$ is translated into $\circledcirc \in \mathcal{E}$.

The encodings of the operands and of the action parameters follow in the sequence the encodings of the corresponding operators and actions, respectively, and are delimited by symbols acting as separators. The set of symbols returned by the encoding contains all these separators. Consider for example the simple PEP system $\diamond \circ \diamond$. The encoding translates it into a CLS sequence composed by a *circ* symbol followed by the encoding of the two operands of $\circ$, namely the two units $\diamond$. A fresh alphabet symbol is used to separate

the three objects, hence we obtain $circ \cdot a \cdot \mathbf{0} \cdot a \cdot \mathbf{0}$ where $a \in \mathcal{E}$ is the separator. In general, terms are considered modulo renaming of its separators. Renaming must not replace separators with other symbols already present in the same sequence.

Moreover, the alphabet symbol *act* is used in the result of the encoding as a program counter: during the evolution of the term it precedes every element which encodes a currently active action. In the definition of the encoding $T\{^y/_x\}$ denotes the substitution in T of each occurrence of x with y.

Definition 19 (Encoding). The encoding of a system P of the PEP calculus into CLS is the term $T \in \mathcal{T}$ such that, for some (finite) $E \subset \mathcal{E}$, it holds that $\{\!|P|\!\} = (T, E)$, where $\{\!|\cdot|\!\} : PEP \to \mathcal{T} \times \mathcal{P}(\mathcal{E})$ is given by the following recursive definition:

$$\{\!|\diamond|\!\} = \big(act \cdot \mathbf{0}, \varnothing\big)$$

$$\{\!|P_1 \circ P_2|\!\} = \big(act \cdot circ \cdot a \cdot P_1'\{^\epsilon/_{act}\} \cdot a \cdot P_2'\{^\epsilon/_{act}\}, \{a\} \cup E_1 \cup E_2\big)$$
$$\text{where } \{\!|P_i|\!\} = (P_i', E_i), E_1 \cap E_2 = \varnothing \text{ and } a \in \mathcal{E} \setminus (E_1 \cup E_2)$$

$$\{\!|!P|\!\} = \big(act \cdot bangS \cdot P'\{^\epsilon/_{act}\}, E\big) \qquad \text{where } \{\!|P|\!\} = (P', E)$$

$$\{\!|\sigma(\!|P|\!)|\!\} = \big(act \cdot brane \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a \cdot P'\{^\epsilon/_{act}\}, \{a\} \cup E_P \cup E_\sigma\big)$$
$$\text{where } \{\!|P|\!\} = (P', E_P), [\![\sigma]\!] = (\sigma', E_\sigma), a \in \mathcal{E} \setminus (E_P \cup E_\sigma) \text{ and } E_P \cap E_\sigma = \varnothing$$

where $[\![\cdot]\!] : Branes \to \mathcal{T} \times \mathcal{P}(\mathcal{E})$ is given by the following recursive definition:

$$[\![\mathbf{0}]\!] = \big(act \cdot \mathbf{0}, \varnothing\big)$$

$$[\![\sigma_1|\sigma_2]\!] = \big(act \cdot par \cdot a \cdot \sigma_1'\{^\epsilon/_{act}\} \cdot a \cdot \sigma_2'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E_1 \cup E_2\big)$$
$$\text{where } [\![\sigma_i]\!] = (\sigma_i', E_i), E_1 \cap E_2 = \varnothing \text{ and } a \in \mathcal{E} \setminus (E_1 \cup E_2)$$

$$[\![!\sigma]\!] = \big(act \cdot bangB \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E\big) \qquad \text{where } [\![\sigma]\!] = (\sigma', E) \text{ and } a \in \mathcal{E} \setminus E$$

$$[\![\phi_n.\sigma]\!] = \big(act \cdot \phi \cdot n \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E\big) \qquad \text{where } [\![\sigma]\!] = (\sigma', E) \text{ and } a \in \mathcal{E} \setminus E$$

$$[\![\phi_n^\perp(\rho).\sigma]\!] = \big(act \cdot \phi^\perp \cdot n \cdot a \cdot \rho'\{^\epsilon/_{act}\} \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E_\rho \cup E_\sigma\big)$$
$$\text{where } [\![\rho]\!] = (\rho', E_\rho), [\![\sigma]\!] = (\sigma', E_\sigma), E_\rho \cap E_\sigma = \varnothing \text{ and } a \in \mathcal{E} \setminus (E_\rho \cup E_\sigma)$$

$$[\![\varepsilon_n.\sigma]\!] = \big(act \cdot \varepsilon \cdot n \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E\big) \qquad \text{where } [\![\sigma]\!] = (\sigma', E) \text{ and } a \in \mathcal{E} \setminus E$$

$$[\![\varepsilon_n^\perp.\sigma]\!] = \big(act \cdot \varepsilon^\perp \cdot n \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E\big) \qquad \text{where } [\![\sigma]\!] = (\sigma', E) \text{ and } a \in \mathcal{E} \setminus E$$

$$[\![\circledcirc(\rho).\sigma]\!] = \big(act \cdot \circledcirc \cdot a \cdot \rho'\{^\epsilon/_{act}\} \cdot a \cdot \sigma'\{^\epsilon/_{act}\} \cdot a, \{a\} \cup E_\rho \cup E_\sigma\big)$$
$$\text{where } [\![\rho]\!] = (\rho', E_\rho), [\![\sigma]\!] = (\sigma', E_\sigma), E_\rho \cap E_\sigma = \varnothing \text{ and } a \in \mathcal{E} \setminus (E_\rho \cup E_\sigma)$$

In Figure 8 we report the rewrite rules which are applicable to encoded PEP systems. Rules are conceptually of two kinds. Rules from rule (par) to rule (sc5) rearrange elementary CLS sequences encoding PEP systems and membranes into CLS terms (containing all CLS operators) and simplifying them according to structural congruence on PEP terms. We denote with $\mathcal{R}_{\langle\rangle}$ this set of rules. Rules from rule (phago) to rule (bangB) correspond to PEP semantics. In particular, rules (phago), (exo) and (pino) correspond to phagocytosis, exocytosis and pinocytosis, respectively, and rules (bangS) and (bangB) correspond to structural congruence for the replication operator. Note that element variables are used repeatedly in rules to match exactly the symbols introduced as separators and identify exactly the subsequences representing the encoding of operands and action parameters.

We remark that by applying rules in $\mathcal{R}_{\langle\rangle}$ to the encoding of a PEP system P we obtain a term T in which each membrane system $(\!|P'|\!)$ in P is represented by a looping sequence in T, and each occurrence of $\circ$ in P is represented by an occurrence of $\mid$ in T. By applying the rules in $\mathcal{R}_{\langle\rangle}$ we obtain the normal form of a PEP system translated into CLS. In [BMM+06a] we have shown that such a normal form is the unique

$$\left(act \cdot par \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z} \cdot x \cdot \widetilde{w}\right)^{L} \rfloor X \;\mapsto\; \left(act \cdot \widetilde{y} \cdot act \cdot \widetilde{z} \cdot \widetilde{w}\right)^{L} \rfloor X \qquad \text{(par)}$$

$$act \cdot circ \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z} \;\mapsto\; act \cdot \widetilde{y} \mid act \cdot \widetilde{z} \qquad \text{(circ)}$$

$$act \cdot brane \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z} \;\mapsto\; \left(act \cdot \widetilde{y}\right)^{L} \rfloor act \cdot \widetilde{z} \qquad \text{(brane)}$$

$$x \cdot \widetilde{w} \mid act \cdot \mathbf{0} \;\mapsto\; x \cdot \widetilde{w} \qquad act \cdot bangS \cdot \mathbf{0} \;\mapsto\; act \cdot \mathbf{0} \qquad \left(act \cdot \mathbf{0}\right)^{L} \rfloor act \cdot \mathbf{0} \;\mapsto\; act \cdot \mathbf{0} \qquad \text{(sc1,2,3)}$$

$$\left(act \cdot \mathbf{0} \cdot x \cdot \widetilde{w}\right)^{L} \rfloor X \;\mapsto\; \left(x \cdot \widetilde{w}\right)^{L} \rfloor X \qquad \left(act \cdot bangB \cdot \mathbf{0} \cdot \widetilde{w}\right)^{L} \rfloor X \;\mapsto\; \left(act \cdot \mathbf{0} \cdot \widetilde{w}\right)^{L} \rfloor X \qquad \text{(sc4,5)}$$

$$\left(act \cdot \phi^{\perp} \cdot x_n \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z} \cdot x \cdot \widetilde{w}\right)^{L} \rfloor X \mid \left(act \cdot \phi \cdot x_n \cdot x' \cdot \widetilde{y}' \cdot x' \cdot \widetilde{z}'\right)^{L} \rfloor Y$$
$$\mapsto\; \left(act \cdot \widetilde{z} \cdot \widetilde{w}\right)^{L} \rfloor \left(X \mid \left(act \cdot \widetilde{y}\right)^{L} \rfloor \left(act \cdot \widetilde{y}' \cdot \widetilde{z}'\right)^{L} \rfloor Y\right) \qquad \text{(phago)}$$

$$\left(act \cdot \varepsilon^{\perp} \cdot x_n \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z}\right)^{L} \rfloor \left(X \mid \left(act \cdot \varepsilon \cdot x_n \cdot x' \cdot \widetilde{y}' \cdot x' \cdot \widetilde{z}'\right)^{L} \rfloor Y\right)$$
$$\mapsto\; Y \mid \left(act \cdot \widetilde{y} \cdot \widetilde{z} \cdot act \cdot \widetilde{y}' \cdot \widetilde{z}'\right)^{L} \rfloor X \qquad \text{(exo)}$$

$$\left(act \cdot \circledcirc \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z} \cdot x \cdot \widetilde{w}\right)^{L} \rfloor X \;\mapsto\; \left(act \cdot \widetilde{z} \cdot \widetilde{w}\right)^{L} \rfloor \left(X \mid \left(act \cdot \widetilde{y}\right)^{L}\right) \qquad \text{(pino)}$$

$$act \cdot bangS \cdot \widetilde{x} \;\mapsto\; act \cdot bangS \cdot \widetilde{x} \mid act \cdot \widetilde{x} \qquad \text{(bangs)}$$

$$\left(act \cdot bangB \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{w}\right)^{L} \rfloor X \;\mapsto\; \left(act \cdot bangB \cdot x \cdot \widetilde{y} \cdot x \cdot act \cdot \widetilde{y} \cdot \widetilde{w}\right)^{L} \rfloor X \qquad \text{(bangb)}$$

Fig. 8. Rewrite rules associated with the encoding of the PEP calculus.

CLS term (modulo structural congruence), denoted $\langle T \rangle$, such that $T \xrightarrow{*} \langle T \rangle$ and $\langle T \rangle \not\rightarrow$. In [BMM+06a] we have also proved that the proposed encoding is sound and complete.

5.2. Comparison between PEP and CLS Bisimilarities

We have defined bisimulation relations for the PEP calculus and for CLS, and we have that the former can be translated into the latter by using the encoding we have given in the previous section. Now, it would be interesting to verify whether there is some relationship between the bisimulation relations of the two formalisms. More precisely, we would like to verify whether the bisimilarity of two PEP systems is preserved by their CLS translations.

We start with comparing strong bisimilarities. It is easy to see that the strong bisimilarity is not preserved by the encoding (i.e. $P \simeq Q \implies \{\!|P|\!\} \sim \{\!|Q|\!\}$ does not hold) because $\Box$–labelled transitions are performed by the encoded systems to create looping sequences and to simulate some axioms of the structural congruence of the PEP calculus. As an example, consider the PEP calculus systems $P = \diamond$ and $Q = \mathbf{0}(\!| \diamond |\!)$. These two systems are structurally congruent, and both perform no transition. However, while the encoding of the former (that is $act \cdot \mathbf{0}$) performs no transition, the encoding of the latter (that is $act \cdot brane \cdot a \cdot \mathbf{0} \cdot a \cdot \mathbf{0}$) performs two $\Box$–labelled transitions, the first caused by the application of the (brane) rule, and the second by the application of the (sc3) rule (the term reached at the end is $act \cdot \mathbf{0}$).

However, we are able to show that the encoding adds only silent behaviour to a PEP system. Namely, the following theorem relates the strong bisimilarity on PEP systems with the weak bisimilarity on CLS terms.

Theorem 7. Given two systems P, Q of the PEP calculus, it holds that: $P \simeq Q \implies \{\!|P|\!\} \approx \{\!|Q|\!\}$.

To prove the theorem we introduce the following three lemmata.

Lemma 6. Given two CLS terms T_1, T_2 obtained from the encoding, it holds that: $\langle T_1 \rangle \equiv \langle T_2 \rangle \implies T_1 \approx T_2$.

Proof. Trivial: the transition system of $\langle T_i \rangle$ is the same as that of T_i apart from the transitions $\xrightarrow{\Box}$ due to the application of rules in $\mathcal{R}_{\langle\rangle}$. □

Lemma 7. Given a PEP system P, it holds that: $\{\!|P|\!\} \approx \langle \{\!|P|\!\} \rangle$.

Proof. Because $\langle \{\!|P|\!\} \rangle \equiv \langle \langle \{\!|P|\!\} \rangle \rangle$, and by Lemma 6. □

Lemma 8. $P \doteq Q \implies \langle \{\!|P|\!\} \rangle \approx \langle \{\!|Q|\!\} \rangle$.

Proof. By definition of $\doteq$ we know that if $P \xrightarrow{\ell} P'$ for some label ℓ, then $Q \xrightarrow{\ell'} Q'$ with $P' \doteq Q'$ and ℓ equivalent to ℓ'. We show that there exist P'' and Q'' such that: (i) the former can be constructed from ℓ and P' and the latter from ℓ' and Q'; (ii) $P'' \doteq Q''$; and (iii) $\langle \{\!|P|\!\} \rangle \overset{C}{\Longrightarrow} \langle \{\!|P''|\!\} \rangle$ and $\langle \{\!|Q|\!\} \rangle \overset{C}{\Longrightarrow} \langle \{\!|Q''|\!\} \rangle$. Since all the transitions performed by $\langle \{\!|P|\!\} \rangle$ and $\langle \{\!|Q|\!\} \rangle$ can be constructed in this manner, we have that $\langle \{\!|P|\!\} \rangle \approx \langle \{\!|Q|\!\} \rangle$.

If the transition performed by both P and Q is a silent transition, namely $\ell = \ell' = \_$, we have that P'' and Q'' correspond to P' and Q', respectively. The same holds when both P and Q perform a $\phi_n^\perp$ action, and in all these cases we have $P'' \doteq Q''$ by definition of the strong bisimulation on PEP systems.

More complex is the case in which both P and Q perform either a ϕ_n or a ε_n action, because they originate, in the corresponding CLS term, a transition for every possible context in which the action can be performed, and each of these transitions leads to a state in which the context has been incorporated in the term. We consider only the case of the ϕ_n action as the case of ε_n is analogous. We have that $\ell = (\phi_n, \sigma(\!|R|\!))$ and $\ell' = (\phi_n, \sigma'(\!|R'|\!))$, with $\sigma(\!|R|\!) \doteq \sigma'(\!|R'|\!)$, and we can construct infinitely many pairs of processes P'' and Q'' for each possible process having the form $\phi_n^\perp(\rho).\tau | \tau_0(\!|R_0|\!)$. More precisely, we have $P'' = \tau | \tau_0(\!|\rho(\!|\sigma(\!|R|\!)|\!) \circ R_0|\!)$ and $Q'' = \tau | \tau_0(\!|\rho(\!|\sigma'(\!|R'|\!)|\!) \circ R_0|\!)$. Since $\sigma(\!|R|\!)$ and $\sigma'(\!|R'|\!)$ are bisimilar and are placed in the same contexts, and since bisimulation is a congruence we have that $P'' \doteq Q''$.

So far we have proved points (i) and (ii). The proof of point (iii) follows from the definition of the rewrite rules associated with the encoding. □

The proof of Theorem 7 follows directly from Lemma 7 and Lemma 8. Moreover, we can give a counterexample to show that the inverse of Theorem 7 does not hold. Consider the PEP systems $P = \diamond$ and $Q = \varepsilon_n^\perp(\!|\varepsilon_n(\!|\diamond|\!)|\!)$. Their encodings are weakly bisimilar as the encoding of the former performs no transition, while the encoding of the latter performs only $\Box$–labelled transition, however the two PEP systems are not strongly bisimilar, as in the PEP labelled semantics, the former performs no transition and the latter one $\_$–labelled transition.

A stronger correspondence exists between the two weak bisimilarity relations of the PEP calculus and CLS. We show that the encodings of two weakly bisimilar PEP systems are two weakly bisimilar terms, and vice versa.

Theorem 8 (Full Abstraction). Given two systems P, Q of the PEP calculus, the following holds:

$$P \fallingdotseq Q \iff \{\!|P|\!\} \approx \{\!|Q|\!\}.$$

Proof. Lemma 7 allow us to reduce the proof of the theorem to the proof of $P \fallingdotseq Q \iff \langle \{\!|P|\!\} \rangle \approx \langle \{\!|Q|\!\} \rangle$. To prove direction $\Longrightarrow$ we first notice that $P \overset{-}{\rightharpoonup} P'$ implies $\langle \{\!|P|\!\} \rangle \overset{\Box}{\Longrightarrow} \langle \{\!|P'|\!\} \rangle$. This can be proved by induction on the derivation of $P \overset{-}{\rightharpoonup} P'$. The base cases are when the last applied rules of the PEP labelled semantics are either (Pi1), or (Ph3), or (E2). In all these three cases we have that $\langle \{\!|P|\!\} \rangle$ can perform a $\Box$–labelled transition caused by the application of one of the CLS rewrite rules associated with the encoding, namely (pino), (phago) and (exo), respectively. After the application of these rules, a state equivalent to $\langle \{\!|P'|\!\} \rangle$ is reached by applying rewrite rules in $\mathcal{R}_{\langle\rangle}$, whose application causes other $\Box$–labelled transitions. The rest of the proof is similar to the proof of Lemma 8.

As regards direction $\Longleftarrow$, we first notice that, by the assumption that we start from terms which are in normal forms, namely $\langle \{\!|P|\!\} \rangle$ and $\langle \{\!|Q|\!\} \rangle$, we have that the only transitions that can be performed are those related to the rewrite rules (phago),(exo),(pino),(bangS) and (bangB). As regards rules (bangS) and (bangB) we have that they cause $\Box$–labelled transitions leading to terms that are the normal forms of the translations of other PEP processes that are structurally congruent to P and Q. Now, the proof consists of showing that if $\langle \{\!|P|\!\} \rangle \xrightarrow{C} T_1$ and $\langle \{\!|Q|\!\} \rangle \xrightarrow{C} T_2$ with $T_1 \approx T_2$, then there exist ℓ, P' and Q' such that $P \xrightarrow{\ell} P'$ and $Q \xrightarrow{\ell} Q'$, and $\langle T_1 \rangle$ and $\langle T_2 \rangle$ are structurally congruent to the translations of PEP processes resulting

from compositions of ℓ and P' and ℓ and Q'. This can be done by cases on the rewrite rules applied to derive the transitions of $\langle\{\![P]\!\}\rangle$ and$\langle\{\![Q]\!\}\rangle$, by analyzing the structure of T_1 and T_2, and by reconstructing ℓ, P' and Q'. This is the inverse procedure of the one used in the proof of Lemma 8 to construct P'' and Q''.

Let us show the case in which the rule (phago) is applied to derive the transition $\langle\{\![P]\!\}\rangle \xrightarrow{C} T_1$. Note that, since the context labeling the transition derived from $\langle\{\![Q]\!\}\rangle$ is the same as the one that appears in the transition performed by $\langle\{\![P]\!\}\rangle$, and since $T_1 \approx T_2$, we know that the same rule (phago) is applied in both the cases of $\langle\{\![P]\!\}\rangle$ and $\langle\{\![Q]\!\}\rangle$. By definition of (phago), the context C can be either $\Box$, or an instantiation of $\Box \mid \left(act \cdot \phi^{\perp} \cdot x_n \cdot x \cdot \widetilde{y} \cdot x \cdot \widetilde{z}\right)^L \rfloor X$, or an instantiation of $\Box \mid \left(act \cdot \phi \cdot x_n \cdot x' \cdot \widetilde{y'} \cdot x' \cdot x' \cdot \widetilde{z'}\right)^L \rfloor Y$. Let us consider the case in which the context has the form $\Box \mid \left(act \cdot \phi \cdot n \cdot a \cdot S_1 \cdot a \cdot S_2\right)^L \rfloor T$. Thus, $\langle\{\![P]\!\}\rangle$ has a form that is structurally congruent to $T_1' \mid \left(act \cdot \phi^{\perp} \cdot n \cdot b \cdot S_1' \cdot b \cdot S_2' \cdot b \cdot S_3'\right)^L \rfloor T_1''$ and $\langle\{\![Q]\!\}\rangle$ has a form that is structurally congruent to $T_2' \mid \left(act \cdot \phi^{\perp} \cdot n \cdot c \cdot S_1'' \cdot c \cdot S_2'' \cdot c \cdot S_3''\right)^L \rfloor T_2''$. By the semantics of CLS, we obtain that T_1 has a form that is structurally congruent to $T_1' \mid \left(act \cdot S_2' \cdot S_3'\right)^L \rfloor (T_1'' \mid \left(act \cdot S_1'\right)^L \rfloor \left(act \cdot S_1 \cdot S_2\right)^L \rfloor T)$ and T_2 has a form that is structurally congruent to $T_2' \mid \left(act \cdot S_2'' \cdot S_3''\right)^L \rfloor (T_2'' \mid \left(act \cdot S_1''\right)^L \rfloor \left(act \cdot S_1 \cdot S_2\right)^L \rfloor T)$.

We can now obtain P and Q from $\langle\{\![P]\!\}\rangle$ and $\langle\{\![Q]\!\}\rangle$, respectively. In particular, we have that P has the form $\phi_n^{\perp}(\rho) \cdot \sigma | \sigma' (\!| P_1 |\!) \circ P_1'$ such that $\langle[\![\rho]\!]\rangle \equiv S_1', \langle[\![\sigma]\!]\rangle \equiv S_2', \langle[\![\sigma']\!]\rangle \equiv S_3', \langle\{\![P_1']\!\}\rangle \equiv T_1'$ and $\langle\{\![P_1]\!\}\rangle \equiv T_1''$. Similarly, we have that Q has the form $\phi_n^{\perp}(\rho') \cdot \tau | \tau' (\!| P_2 |\!) \circ P_2'$ such that $\langle[\![\rho']\!]\rangle \equiv S_1'', \langle[\![\tau]\!]\rangle \equiv S_2'', \langle[\![\tau']\!]\rangle \equiv S_3'', \langle\{\![P_2']\!\}\rangle \equiv T_2'$ and $\langle\{\![P_2]\!\}\rangle \equiv T_2''$. By the semantics of the PEP calculus, we have that both P and Q can perform a transition with $\ell = (\phi_n^{\perp}, \phi_n.\sigma'' | \tau'' (\!| R |\!)$ as label, where $S_1 \equiv \langle[\![\sigma'']\!]\rangle, S_2 \equiv \langle[\![\tau'']\!]\rangle$ and $T \equiv \langle\{\![R]\!\}\rangle$. Thus, P' and Q' are the PEP systems $\sigma | \sigma' (\!| \rho (\!| \sigma'' | \tau'' (\!| R |\!) |\!) |\!)$ and $\tau | \tau' (\!| \rho' (\!| \sigma'' | \tau'' (\!| R |\!) |\!) |\!)$, respectively. □

6. Discussion

In this paper we have defined semantics based on labelled transition systems and bisimulations for two calculi for the description of biological systems.

The first calculus we have considered is the Calculus of Looping Sequences (CLS), that is based on term rewriting. In order to define the labelled semantics we have followed the approach of using contexts as labels, which has been proposed by Sewell [Sew02] and by Leifer and Milner [LeM00]. In such an approach the contexts used as labels of the transitions performed from a state of the system are derived from the rewrite rules, and this allows defining transition systems for which bisimilarities are congruences.

The second calculus we have considered is the PEP calculus, which is the simplest of Brane Calculi and is a representative of those languages for the description of biological systems that fall in the category of process calculi. In order to deal with the actions of the PEP calculus having membranes as parameters, we have followed a strategy for the definition of the labelled semantics that is commonly used for high–order process calculi [San96], namely we have used pairs as labels in which the first element is the action performed by the transition and the second element is the membrane used as parameter of the action.

To define semantics and bisimulations of CLS and the PEP calculus we have followed two approaches, each typical of the class to which the considered language belongs. The result we have proved (that by using the encoding of the PEP calculus into CLS, equivalent PEP systems correspond to equivalent CLS terms) suggests that both languages are able to properly describe the behaviour of the considered biological systems and that there exists a meaningful notion of equivalence for the behaviour of such systems.

The semantics we have defined, both for CLS and PEP, allow the observation of the context in which the described systems evolve. In particular, in the semantics of CLS what is observable is exactly the context in which a term could evolve, while in the semantics of PEP what is observable are the processes representing such contexts. A characteristic of biological systems is indeed that they evolve under stimuli from an environment. Consider as examples proteins which undergo transformations due to the presence of enzymes, and signalling pathways which are triggered by signal proteins. Therefore, semantics based on the mentioned concepts of observation seem to be suitable to model biological systems.

We have presented two case studies of biological systems in which bisimulations have been used to verify a causality property, in the first case, and to compare two behaviours arising in a system due to different causes, in the second case. The property proved in the first case could have been proved by model checking as well, while performing the comparison of the second case by model checking would not be easy.

We mention that bisimulations could be used to define operations on the model which reduce the model

itself while preserving all properties which can be expressed in a temporal logic, as done in [AMP+04] where hybrid automata are used to describe systems and bisimulation is exploited in the definition of a projection operator that makes possible to work with reduced automata. A similar approach is followed in [PFM+07], where bisimulation congruences of CCS [Mil89] are used to compositionally reduce the size of a model of the lactose operon regulation in order to facilitate verification of properties of the system by model checking. Bisimulations for the analysis of the evolution of biological systems have been considered also in [LaT06] where they are defined for an extension of the κ–calculus [DaL04] with the aim of comparing the behaviour of systems. These bisimulations are proved to be congruences.

Other behavioural equivalences could be considered rather than bisimulations. Some among the most commonly used are trace equivalences [vGl90], barbed congruences [MiS92] and testing equivalences [DeH84]. Trace equivalences compare the set of sequences of labels (traces) of two labelled transition systems. Traces are obtained by concatenating the labels of the transitions of each possible execution path. These equivalences are typically coarser than bisimulations and could be defined for the semantics we have introduced to compare the behaviour of systems without taking branching into account. Even if they are coarser than bisimulations, trace equivalences could be fine enough to compare the behaviours and to prove properties of many relevant biological systems. Barbed congruences can be defined for transition systems without labels (such as the standard CLS and PEP semantics) and require the definition of an observation relation on states. Such a relation typically describes the interactions with the environment that are enabled in each state. The definition of barbed congruences for CLS and the PEP calculus could lead to equivalences similar (if not identical) to our bisimulation relations. Finally, testing equivalences are typically defined for process calculi by means of observer processes which have the same syntax of the processes to be compared and can perform a special action ω (the success action). The idea is that two processes are equivalent if they interact similarly with all the observer processes composed in parallel with each of them. In particular, they are equivalent if for each observer either both or neither of them can perform a computation leading to the execution of ω (may–testing) or are such that all computations lead to the execution of ω (must–testing). Also testing equivalences could be applied to descriptions of biological systems, but they could be hard to verify because of the universal quantification over observer processes in their definition.

Verification of properties for systems described as Brane Calculi is tackled in [Bus06, Bus07] and [MiB06]. In [Bus06] the decidability of divergence, control state maintainability, inevitability and boundedness properties is proved for the calculus of the family with molecules and without the phagocytosis action. In [Bus07] a semantics is proposed for the Mate/Bud/Drip (MBD) brane calculus describing the causal dependencies occurring between the reactions of a system. The primitives of MBD are inspired by membrane fusion (Mate) and fission (Bud and Drip). The causal and labelled semantics in [Bus07] decorates the reaction relation with a fresh name associated to the given reaction and a set containing the names of the reactions that already occurred (thus representing a cause for the current reaction). In this paper, the relevance of causality for biological systems is put in evidence showing which events, occurring in a biological pathway, are necessary for other events to happen. In [MiB06] the Brane Logic is introduced together with a proof system and a model checker for a decidable fragment of the logic.

As future work, on the one hand we plan to continue the theoretical investigation of behavioural equivalences for formalisms devoted to the modeling of biological systems. In particular, we are interested in considering the above mentioned equivalences, compare them and discuss their applicability to systems biology. On the other hand, we plan to apply behavioral equivalences to the study of some relevant biological problems.

Acknowledgments

This research has been partially supported by MiUR PRIN 2006 Project “Biologically Inspired Systems and Calculi and their Applications (BISCA)”.

References

[ABI+01] Alur, R., Belta, C., Ivancic, F., Kumar, V., Mintz, M., Pappas, G. J., Rubin, H. and Schug, J.: Hybrid Modeling and Simulation of Biomolecular Networks. *Proc. of Hybrid Systems: Computation and Control*, LNCS 2034, pp. 19–32, Springer, 2001.

[AMP+04] Antoniotti, M., Mishra, Piazza, C., Policriti, A. and Simeoni, M.: Taming the Complexity of Biochemical Models through Bisimulation and Collapsing: Theory and Practice. *Theoretical Computer Science*, **325**, pp. 46–67, 2004.

[BMM+06a] Barbuti, R.,, Maggiolo-Schettini, A., Milazzo, P. and Troina, A. : A Calculus of Looping Sequences for Modelling Microbiological Systems. *Fundamenta Informaticae* **72**, pp. 1–15, 2006.

[BMM+06b] Barbuti, R.,, Maggiolo-Schettini, A., Milazzo, P. and Troina, A. : Bisimulation Congruences in the Calculus of Looping Sequences. *Proc. of International Colloquium on Theoretical Computer Science (ICTAC'06)*, LNCS 4281, pp. 93–107, Springer, 2006.

[Bus06] Busi, N.: Deciding Behavioural Properties in Brane Calculi *Proc. of Computational Methods in Systems Biology (CMSB'06)*, LNBI 4210, pp. 17–31, Springer, 2006.

[Bus07] Busi, N.: Towards a Causal Semantics for Brane Calculi *Proc. of the Fifth Brainstorming Week on Membrane Computing*, pp. 97–111, 2007.

[Car05] Cardelli, L.: Brane Calculi. Interactions of Biological Membranes. *Proc. of Computational Methods in Systems Biology (CMSB'04)*, LNCS 3082, pp. 257–280, Springer, 2005.

[CCD04] Chabrier–Rivier, N., Chiaverini, M., Danos, V., Fages, F. and Schachter, V.: Modeling and Querying Biomolecular Interaction Networks. *Theoretical Computer Science* **325**, pp. 25-44, 2004.

[CDP+04] Curti, M., Degano, P., Priami, C. and Baldari, C. T.: Modelling Biochemical Pathways through Enhanced pi-calculus. *Theoretical Computer Science* **325**, pp. 111–140, 2004.

[DaL04] Danos, V. and Laneve, C.: Formal Molecular Biology. *Theoretical Computer Science* **325**, pp. 69–110, 2004.

[DeH84] De Nicola, R. and Hennessy, M.: Testing Equivalences for Processes. *Theoretical Computer Science* **34**, pp. 83–133, 1984.

[LaT06] Laneve, C. and Tarissan, F.: A Simple Calculus for Proteins and Cells. *Proc. of Membrane Computing and Biologically Inspired Process Calculi (MeCBIC'06)*, 2006.

[LeM00] Leifer, J. and Milner, R.: Deriving Bisimulation Congruences for Reactive Systems. *Proc. of Concurrency Theory (CONCUR'00)*, LNCS 1877, pp. 243–258, Springer, 2000.

[MDN+00] Matsuno, H., Doi, A., Nagasaki, M. and Miyano, S.: Hybrid Petri Net Representation of Gene Regulatory Network. *Proc. of Pacific Symposium on Biocomputing*, pp. 341–352, World Scientific Press, 2000.

[MiB06] Miculan, M. and Bacci, G.: Modal Logics for Brane Calculus *Proc. of Computational Methods in Systems Biology (CMSB'06)*, LNBI 4210, pp. 1–16, Springer, 2006.

[Mil07] Milazzo, P.: Qualitative and Quantitative Formal Modeling of Biological Systems, PhD Thesis, University of Pisa, 2007.

[Mil89] Milner, R.: *Communication and Concurrency*, Prentice–Hall, 1989.

[Mil99] Milner, R.: *Communicating and Mobile Systems: the π–Calculus*, Cambridge University Press, 1999.

[MiS92] Milner, R and Sangiorgi, D.: Barbed Bisimulation. *Proc. of Int. Colloquium on Automata, Languages and Programming (ICALP'92)*, LNCS 623, pp. 685–695, 1992.

[PFM+07] Pinto, M.C., Foss, F., Mombach, J.C.M. and Ribeiro, L.: Modelling, Property Verification and Behavioural Equivalence of Lactose Operon Regulation. *Computer in Biology and Medicine* **37**, pp. 134–148, 2007.

[PrQ05] Priami, C. and Quaglia, P.: Beta Binders for Biological Interactions. *Proc. of Computational Methods in Systems Biology (CMSB'04)*, LNCS 3082, pp. 20–33, Springer, 2005.

[RPS+04] Regev, A., Panina, E. M., Silverman, W., Cardelli, L. and Shapiro, E.: BioAmbients: An Abstraction for Biological Compartments. *Theoretical Computer Science* **325**, pp. 141–167, 2004.

[RSS01] Regev, A., Silverman, W. and Shapiro, E. Y.: Representation and Simulation of Biochemical Processes Using the pi-calculus Process Algebra. *Proc. of Pacific Symposium on Biocomputing*, pp. 459–470, World Scientific Press, 2001.

[San96] Sangiorgi, D.: Bisimulation for Higher–Order Process Calculi. *Information and Computation* **131**, pp. 141–178, 1996

[Sew02] Sewell, P.: From Rewrite Rules to Bisimulation Congruences. *Theoretical Computer Science* **274**, pp. 183–230, 2002.

[SHJ+93] Straight, S.W., Hinkle, P.M., Jewers, R.J., and McCance, D.J.: The E5 oncoprotein of human papillomavirus type 16 transforms fibroblasts and effects the downregulation of the epidermal growth factor receptor in keratinocytes. *Journal of Virology* **67**, pp. 4521–4532, 1993.

[Yar01] Yarden, Y.: The EGFR family and its ligands in human cancer: signalling mechanism and therapeutic opportunities. *European Journal of Cancer* **37**, pp. S3–S8, 2001.

[YS01] Yarden, Y., Sliwkowski, M.K.: Untangling the ErbB signalling network. *Nature Reviews in Molecular Cell Biology* **2**, pp. 127-37, 2001.

[vGl90] van Glabbeek, R.J.: The Linear Time–Branching Time Spectrum. *Proc. of Int. Conference on Concurrency Theory (CONCUR'90)*, LNCS 458, pp. 278–297, 1990.